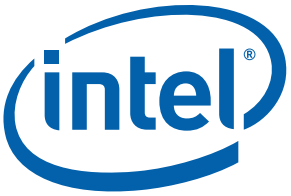


Intel® Atom™ Processor C2000 Product Family for Microserver

Datasheet

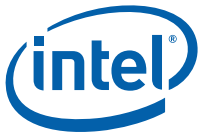
January 2016

Order Number: 330061-003US

Revision History

Date	Revision	Description
January 2016	003US	The following technical changes were made in Datasheet 003US: • Table 1-2 - Added "Lowest Frequency Mode" row to table and removed "Reliability/ Availability" row from table. • Table 1-4 - Added CPPM to terminology. • Section 3.3.1 and Section 3.3.2 - Corrected minimum memory capacity. and device density. • Table 3-1 and Table 3-2 - Added row for 1 GB. • Section 3.4.2 - Added Paragraph. • Section 4.4 - Updated section. • Table 5-2 - Updated signals. • Section 7 - Added note to SoC Reset and Power Supply Sequences. • Section 7.1.3 - Changed sequence shown in V1P0A voltage is provided to all V1P0A voltage-group pins the of SoC. • Section 7.2.1 - Updated Cold Reset Sequence. • Section 12.3.6 - Added note. • Section 15.4.7.1 and Section 15.4.8.1- Updated. • Table 22-3 - Updated entries in Content column. • Section 22.7 - Added note. • Table 23-2 - Updated descriptions. • Table 25-1 - Updated and clarified reserved bits in Section 25.3.1. • Figure 31-1 and Table 31-14 - Updated signal names. • Table 31-15 - Updated RTC Well Signals. • Table 31-17 - Updated Description of PMU_RESETBUTTON_B/GPIOS_30. • Table 31-24 - Updated entries in "Internal Pull-up or Pull-down" column. • Table 31-25 - Updated entries in "Internal Pull-up or Pull-down" column. • Table 31-25 - Added footnote to SPI_CS0_B. • Table 33-2 and Table 33-3 - Updated signal name of T_{DQS_CK}. • Table 33-42 - Updated RTC Crystal Requirements. • Figure 33-10 - Figure updated.
September 2014	002US	The following technical changes were made in Datasheet 002US: • Updated Revision numbering scheme for public release from 1.1 to 002US. • Global Change - IRERR changed to IERR throughout the manual. • Global Change - SMBALERT# changed to SMBALRT_N throughout the manual. • Table 1-2 - Added CUNIT_REG_DEVICEID[31:0] row. • Section 3.3.3 - Added System Memory Technology which is Not Supported. • Table 3-2 - Added Table note. • Section 7.2.1 - Updated V1P35S note. • Section 9.3 - Updated PCI Express. • Section 11.5.2.1 - Added paragraph. • Table 12-2 - Updated Description for FLEX_CLK_SE1. • Section 12.2 - Updated supported features. • Section 12.3.3.2 - Updated ASPM and ASPM Optionality. • Section 12.6 - Updated Power Management. • Table 12-9 - Added x4 Lanes with 4 Controllers SKUs. • Previous Section 16.5.2 - Deleted section. • Table 16-1 - Changed Strap Usage for FLEX_CLK_SE1. • Table 16-5 - Updated Description for Bits 1 and 0. • Table 17-4 - Added Table Note 1 to Slave Address (Data Phase). • Table 31-6 - Updated Description for THERMTRIP_N. • Table 31-24 - Changed FLEX_CLK_SE1 pin 0 to Reserved. • Section 33.16.4 - Changed PPM Tolerance to 35 ppm. • Table 33-45 - Changed TRISE/FALL Min and Max Rise/Fall Time Max from 5ns to 3ns.
January 2014	1.0	Initial Release.

Contents



Figures

Tables

§ §

Volume 1: C2000 Product Family Program Overview

January 2016
Order Number: 330061-003US

1 Introduction and Product Offerings

1.1 Overview

A new emerging segment exists in the server marketplace that Intel Corporation calls Microserver. The Intel® Atom™ Processor C2000 Product Family for Microserver is the next generation of System-On-Chip (SoC) 64-bit processor built on the 22-nanometer process technology designed for certain lightweight scale out workloads. This highly-integrated SoC contains two, four, or eight processor cores depending on the product SKU. Throughout this document, the Intel® Atom™ Processor C2000 Product Family for Microserver is also referred to as simply SoC. This document relates to the product SKUs shown in Table 1-1.

Table 1-1. Intel® Atom™ Processor C2000 Product Family for Microserver (C2xx0)

SKU Name/Number	SKU Description
Intel® Atom™ Processor C2350	2 Core, 1.7 GHz
Intel® Atom™ Processor C2530	4 Core, 1.7 GHz
Intel® Atom™ Processor C2550	4 Core, 2.4 GHz
Intel® Atom™ Processor C2730	8 Core, 1.7 GHz
Intel® Atom™ Processor C2750	8 Core, 2.4 GHz

This market is served by the low-power Intel® Xeon® processor-based systems and recently by platforms based on the Intel® Atom™ processor.

This SoC is the first silicon development to specifically address the Microserver market. As such, the SoC achieves dramatic improvements over past microservers in the five dimensions described above.

Each execution core has an L1 instruction cache and data cache. The execution cores are grouped into two-core modules and share a 1 MB L2 cache and common interface to the rest of the SoC. Each core is a single-threaded execution core and performs out-of-order instruction execution.

The target audience for this document is primarily system architects and board designers who are planning to develop a SoC-based Microserver solution. Additionally, this document is also used by other system engineers such as system test engineers, software developers, and BIOS developers.

1.2 Key Features

The main SoC architectural features are:

- SKUs containing two, four, or eight cores (Table 1-1)
  - Intel® Xeon® processor Instruction Set Architecture (ISA) compatibility
  - Out-of-order instruction execution
  - Intel® Virtualization Technology, VT-x2
  - 1 MB shared L2 Cache (per two cores), 4 MB L2 total for the eight-core SKUs
  - SKU base frequencies of 1.7 GHz and 2.4 GHz
  - Intel® Turbo Boost Technology for speeds up to 2.7 GHz depending on SKU
- Dual-Channel DDR3 Memory
  - Single- or Dual-Channel Memory Controller, SKU dependent
  - DDR3L (1.35V), DDR3 (1.5V), SKU dependency
  - Speeds up to 1600 MT/s depending on SKU
  - ECC support
  - Support for single- or dual-rank DIMMs
  - Support up to two DIMMs per channel
  - Up to 64 GB DDR3 memory capacity support, depending on product SKU.
- Integrated
  - PCI Express* Gen 2 Root Ports, up to 16 lanes, bifurcates to 4 controllers
  - Four Enterprise Class Gigabit Ethernet (GbE) Ports Per SoC (1 Gb or 2.5 Gb)
    - Network Controller Sideband Interface (NC-SI) allows for connectivity to a Baseboard Management Controller (BMC) for the purpose of enabling out-of-band remote manageability.
    - SMBus ports are available and continue to be available to enable network manageability implementations.
    - SoC designs can use either SMBus or NC-SI for connectivity to the BMC, but not both.
    - The SoC GbE Interface provides support for IEEE* 802.3 1000BASE-KX and 2500BASE-X.
  - Four SATA2 Ports to support high-capacity rotational media, one SKU with no SATA2
  - Two SATA3 Ports to support solid-state drives (SSDs) requiring high rates of I/O operations per second (IOPS)
  - USB 2.0 x4, EHCI compliant
  - SMBus x4 (Host, PECI, normal SPD, and LAN interface)
- Based on new Intel SoC design technology
  - Next-Generation SoC System Agent (SSA)
  - Significant improvements in performance and latency than current Intel® Atom™ processor system agents.
  - Common legacy block controllers (SPI, UART, RTC, HPET, etc.)

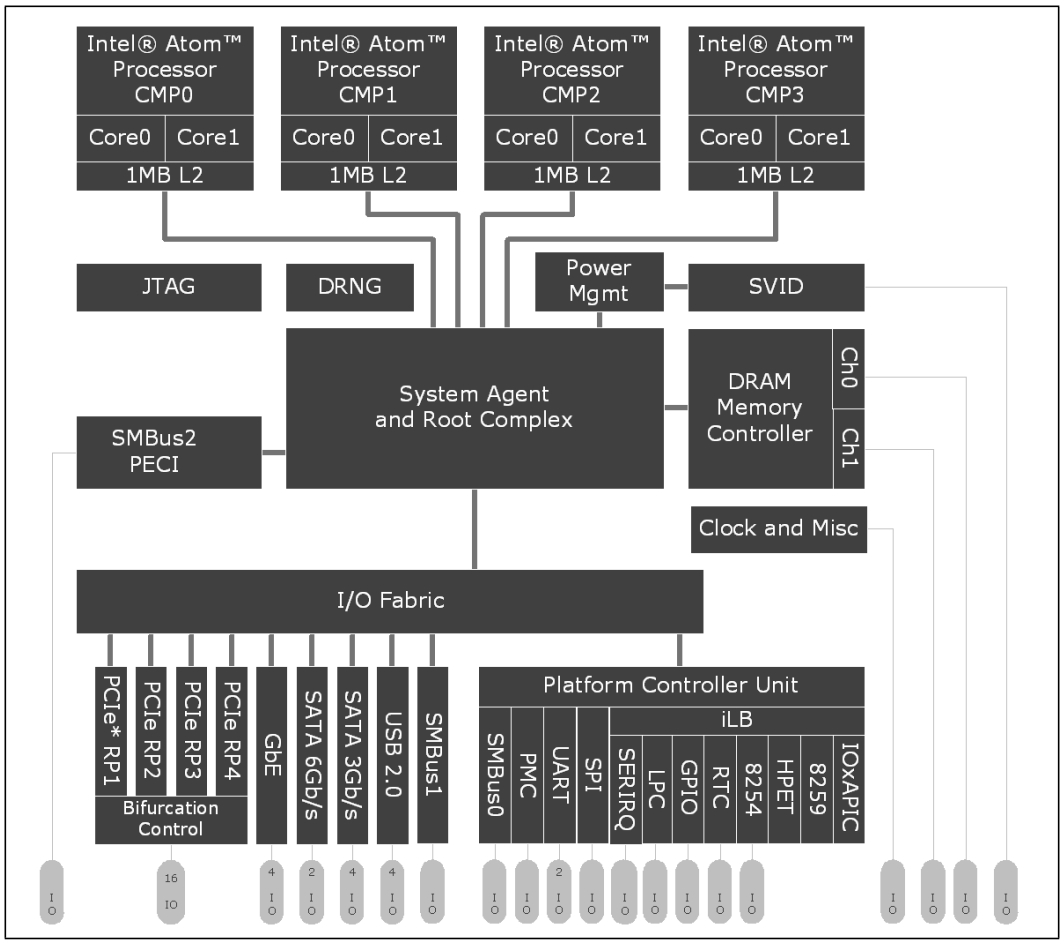
- Power Management
  - — Significant improvements to support lightweight server power management
  - — Exposed PECI over SMBus mechanism
  - — Highly-optimized Power Management Unit (PMU)
  - — Support for Turbo, Running Average Power Limiting (RAPL)
  - — SVID support to optimize power consumption
- Server-Class Reliability, Availability and Serviceability (RAS)
  - — Data and address for memory ECC
  - — Demand and Patrol Scrub to detect and correct memory errors
  - — Significant internal data-path parity protection

Functional descriptions of these SoC features and other integrated devices are in the remaining chapters.

1.3 Intel® Atom™ Processor C2000 Product Family for Microserver Block Diagram

Figure 1-1 shows a high-level SoC block diagram. Maximum configurations shown do not apply to all product SKUs.

Figure 1-1. Intel® Atom™ Processor C2000 Product Family for Microserver High-Level Block Diagram



1.4 Product SKUs

Table 1-2. Intel® Atom™ Processor C2000 Product Family for Microserver Product SKUs

Characteristic	C2750 (SKU 3)	C2730 (SKU 4)	C2550 (SKU 6)	C2530 (SKU 7)	C2350 (SKU 8)
Thermal Design Power (TDP) @$T_{J\text{-}MAX}$, 100°C (Watts)	20W	12W	14W	9W	6W
Core Frequency (GHz)	2.4	1.7	2.4	1.7	1.7
Lowest Frequency Mode	1.2	1.2	1.2	1.2	1.2
Intel®Atom™ Cores	8	8	4	4	2
Intel® Turbo Boost Technology (Max Turbo - aka 1 core Turbo)	2.6	2.4	2.6	2.4	2.0
Intel® Turbo Boost Technology (ALL Core Turbo)	2.6	2.0	2.6	2.0	2.0
DDR3 Memory Channels	2	2	2	2	1
DDR3 DIMMs per Channel	2	2	2	2	2
DDR3 Memory Types Supported	DDR3, DDR3L	DDR3, DDR3L	DDR3, DDR3L	DDR3, DDR3L	DDR3, DDR3L
ECC Supported	Yes	Yes	Yes	Yes	Yes
Non-ECC Supported	UDIMM Only	UDIMM Only	UDIMM Only	UDIMM Only	UDIMM Only
Memory Frequency - Maximum	1600	1600	1600	1333	1333
Memory Capacity - Maximum	64 GB	32 GB	64 GB	32 GB	16 GB
PCIe* Lanes Available	16	8	16	8	4
PCIe Controllers	4	2	4	2	1
GbE Ports	4	2	4	2	4
2.5 GbE	Yes	Yes	Yes	Yes	Yes
SATA3 Ports	2	2	2	2	2
SATA2 Ports	4	0	4	0	0
USB 2.0 Ports	4	4	4	4	4
UART Interfaces	2	2	2	2	2
CPU Virtualization (VT-x2)	Yes	Yes	Yes	Yes	Yes
CUNIT_REG_DEVICED[31:0]	1F01_8086	1F00_8086	1F02_8086	1F03_8086	1F04_8086

Notes:
1. The SKUs with the I/O ports disabled require the power and ground pins connected. The input and output signals may be left as No Connects (NC).
2. The disabled PCIe* lanes operation is discussed in this document. See Section 12.8.2, "PCI Express Lanes with Various SKUs Design Consideration" on page 235 for the per SKU connectivity.

The C2350 (SKU 8) disables the DDR memory channel 1. Even though channel 1 is not used, all channel 1 VDDQ power pins must be supplied power from the VDDQ channel 0 power source. Also, the DDR3 reference-clock differential input pins must both be tied to VSS on the platform board. All other memory signals are not connected (NC).

1.5 Datasheet Volume Structure and Scope

This document is intended to be distributed as a part of the complete Datasheet document, which consists of three volumes. The Datasheet volume structure and scope are provided in Table 1-3.

Table 1-3. Datasheet Volume Structure and Scope (Sheet 1 of 2)

Volume One: SoC Program Overview
• Chapter 1, "Introduction and Product Offerings"
• Chapter 2, "Multi-Core Intel® Atom™ Processors"
Volume Two: Functional Description
• Chapter 3, "Memory Controller"
• Chapter 4, "System Agent and Root Complex"
• Chapter 5, "Clock Architecture"
• Chapter 6, "Interrupt Architecture"
• Chapter 7, "SoC Reset and Power Supply Sequences"
• Chapter 8, "Thermal Management"
• Chapter 9, "Power Management"
• Chapter 10, "System Address Maps"
• Chapter 11, "Gigabit Ethernet (GbE) Controller"
• Chapter 12, "PCI Express Root Ports (RP)"
• Chapter 13, "SATA Controllers (SATA2, SATA3)"
• Chapter 14, "Universal Serial Bus (USB) 2.0"
• Chapter 15, "SMBus 2.0 Unit 1 - Host"
• Chapter 16, "Platform Controller Unit (PCU)"
• Chapter 17, "SMBus 2.0 Unit 2 - PECI"
• Chapter 18, "SMBus 2.0 Unit 0 - PCU"
• Chapter 19, "Power Management Controller (PMC)"
• Chapter 20, "UART Controller"
• Chapter 21, "Intel Legacy Block (iLB) Devices"
• Chapter 22, "Serial Peripheral Interface (SPI)"
• Chapter 23, "Serial Interrupt Controller"
• Chapter 24, "Low Pin Count (LPC) Controller"
• Chapter 25, "General-Purpose I/O (GPIO)"
• Chapter 26, "Real Time Clock (RTC)"
• Chapter 27, "8254 Programmable Interval Timer (PIT)"
• Chapter 28, "High Precision Event Timer (HPET)"
• Chapter 29, "8259 Programmable Interrupt Controller (PIC)"
• Chapter 30, "I/O Advanced APIC (I/O APIC)"

Table 1-3. Datasheet Volume Structure and Scope (Sheet 2 of 2)

Volume Three: Electrical, Mechanical and Thermal Specification
• Chapter 31, “Signal Names and Descriptions”
• Chapter 32, “Signal Pin States and Termination”
• Chapter 33, “Signal Electrical and Timing Characteristics”
• Chapter 34, “Operating Conditions and Power Requirements”
• Chapter 35, “Component Ball-Out Listing”
• Chapter 36, “Mechanical Characteristics”

1.6 Terminology

Table 1-4. Processor Terminology (Sheet 1 of 5)

Term	Description
1000BASE-KX	1000BASE-KX is the IEEE* 802.3ap electrical specification for transmission of 1 Gb/s Ethernet over the backplane.
1000BASE-T	1000BASE-T is the specification for 1 Gb/s Ethernet over category 5e twisted pair cables as defined in IEEE 802.3 clause 40.
2500BASE-X	The 2500BASE-X is link mode is used for Ethernet-over-backplane implementations. The 2500BASE-X link mode is a special, enhanced speed mode of the 1000BASE-X link mode.
2.5 GbE	The 2.5-GbE link mode is a special enhanced-speed mode of the 1000BASE-X link mode. This mode should not be confused with third-party solutions being offered in the marketplace today. Contact the local Intel representative for further information on the scope of this feature.
ACPI	Advanced Configuration and Power Interface
AEN	Asynchronous Event Notification
AFE	Analog Front End
Aggressor	A network that transmits a coupled signal to another network.
Anti-Etch	Any plane-split, void, or cutout in a VCC or GND plane is referred to as an anti-etch.
ARP	Address Resolution Protocol
ASF	Alert Standard Format
b/w	Bandwidth
BER	Bit Error Rate
BGA	Ball Grid Array
BIOS	Basic Input/Output System
BIST FIS	Built-In Self-Test Frame Information Structure
BMC	Baseboard Management Controller (often used interchangeably with MC)
BT	Bit Time
Bus Agent	A component which represents a load on a bus.
CMC	Common Mode Choke
CPPM	Combined Pulse Position Modulation
CRC	Cyclic Redundancy Check
Crosstalk	The reception on a victim network of a signal imposed by aggressor network(s) through inductive and capacitive coupling between the networks. Backward Crosstalk – Coupling that creates a signal in a victim network that travels in the opposite direction as the aggressor signal. Forward Crosstalk – Coupling that creates a signal in a victim network that travels in the same direction as the aggressor signal. Even Mode Crosstalk – Coupling from a signal or multiple aggressors when all the aggressors switch in the same direction that the victim is switching. Odd Mode Crosstalk – Coupling from a signal or multiple aggressors when all the aggressors switch in the opposite direction that the victim is switching.
CSR	Configuration and Status Register
DCA	Direct Cache Access
DDR3	Third generation Double Data Rate SDRAM memory technology that is the successor to DDR2 SDRAM.
DFT	Design for Testability
DMA	Direct Memory Access
DMTF	Distributed Management Task Force standard body

Table 1-4. Processor Terminology (Sheet 2 of 5)

Term	Description
DP	Display Port
DQ	Descriptor Queue
DUT	Device Under Test
DW	Double Word (4 bytes)
ECC	Error Correction Code
EEE	Energy Efficient Ethernet - IEEE 802.3az standard
EEPROM	Electrically Erasable Programmable Memory. A non-volatile memory directly accessible from the host.
EHCI	Enhanced Host Controller Interface
EMI	Electromagnetic Interference
Enhanced Intel SpeedStep® Technology	Allows the operating system to reduce power consumption when performance is not needed.
EOP	End of Packet
ESD	Electrostatic Discharge
Execute Disable Bit	The Execute Disable bit allows memory to be marked as executable or non-executable, when combined with a supporting operating system. If code attempts to run in non-executable memory the processor raises an error to the operating system. This feature prevents some classes of viruses or worms that exploit buffer overrun vulnerabilities and improves the overall system security. See the Intel® 64 and IA-32 architectures software developer manuals for more detailed information.
FC	Flow Control
FCS	Frame Check Sequence
Firmware (FW)	Embedded code on the LAN controller that implements the NC-SI protocol and pass-through functionality.
Flight Time	A term in the timing equation that includes the signal propagation delay, any effects the system has on the driver TCO, plus any adjustments to the signal at the receiver needed to guarantee the receiver setup time. More precisely, flight time is defined as the following: • The time difference between a signal at the input pin of a receiving agent crossing the switching voltage (adjusted to meet the receiver manufacturer conditions required for AC timing specifications; i.e., ringback, etc.) and the output pin of the driving agent crossing the switching voltage when the driver is driving a test load used to specify the driver AC timings. Maximum and Minimum Flight Time – Flight time variations are caused by many different parameters. The more obvious causes include the board dielectric constant variation, changes in load condition, crosstalk, power noise, variation in termination resistance, and differences in I/O buffer performance as a function of temperature, voltage, and manufacturing process. Some less obvious causes include effects of Simultaneous Switching Output (SSO) and packaging effects. Maximum flight time is the largest acceptable flight time a network experiences under all conditions. Minimum flight time is the smallest acceptable flight time a network experiences under all conditions.
FS	Full-speed. Refers to USB.
Host Interface	RAM on the LAN controller shared between the firmware and the host, and used to pass commands from the host to the firmware and responses from the firmware to the host.
HPC	High - Performance Computing
HS	High-speed. Refers to USB.
IIO	The Integrated I/O Controller. An I/O controller that is integrated in the processor die.
iMC	The Integrated Memory Controller. A Memory Controller that is integrated in the processor die.
Intel® 64 Technology	64-bit memory extensions to the IA-32 architecture. Further details on the Intel 64 architecture and the programming model are found at: http://developer.intel.com/technology/intel64/.
IPC	Inter-Processor Communication
IPG	Inter-Packet Gap
IPMI	Intelligent Platform Management Interface specification

Table 1-4. Processor Terminology (Sheet 3 of 5)

Term	Description
Intel® Virtualization Technology (Intel® VT)	Processor virtualization which when used in conjunction with the Virtual Machine Monitor software enables multiple, robust independent software environments inside a single platform.
Integrated Heat Spreader (IHS)	A component of the processor package used to enhance the thermal performance of the package. Component thermal solutions interface with the processor at the IHS surface.
ISI	The effect of a previous signal (or transition) on the interconnect delay. For example, when a signal is transmitted down a line and the reflections due to the transition have not completely dissipated, the following data transition launched onto the bus is affected. Interconnect Symbolic Interference (ISI) is dependent upon frequency, time delay of the line, and the reflection coefficient at the driver and receiver. ISI impacts both timing and signal integrity.
Jitter	Any timing variation of a transition edge or edges from the defined Unit Interval (UI).
LAN (Auxiliary Power-Up)	If connecting the LAN controller to a power source (occurs even before system power-up).
Land	The contact point of a processor in the 559–Ball BGA package.
LCD	Name for the external LAN Connected Device, sometimes also called a PHY device. The LCD name distinguishes between the whole device and the PHY portion within the LCD.
LCI	LAN Connect Interface
LLC	Last Level Cache
LLDP	Link Layer Discovery Protocol defined in IEEE 802.1AB and used by IEEE 802.3az (EEE) for system wake time negotiation.
LOM	LAN on Motherboard
LPC	Low Pin Count
LPI	Low-Power Idle is the low-power state of Ethernet link as defined in IEEE 802.3az.
LRU	Least Recently Used. A term used in conjunction with cache hierarchy.
LS	Low-speed. Refers to USB.
LSO	Large Send Offload
LTR	Latency Tolerance Reporting (PCI Express* (PCIe*) protocol)
LVR	Linear Voltage Regulator
MAC	Media Access Control
MC	Management Controller
MCTP	DMTF Management Component Transport Protocol (MCTP) specification. A transport protocol to allow communication between a management controller and controlled device over various transports.
MDIO	Management Data Input/Output Interface over MDC/MDIO lines
MIFS/MIPG	Minimum Inter-Frame Spacing/Minimum Inter-Packet Gap
MLC	Mid-Level Cache
MMC	Multi-Node Management Controller
MMW	Maximum Memory Window
MPS	Maximum Payload Size in PCIe specification
MSS	Maximum Segment Size. Largest amount of data, in a packet (without headers) that can be transmitted. Specified in bytes.
MTU	Maximum Transmit Unit. Largest packet size (headers and data) that can be transmitted. Specified in bytes.
NC	Network Controller
NC-SI	Network Controller Sideband Interface DMTF Specification
Network	The trace of a Printed Circuit Board (PCB) that completes an electrical connection between two or more components.
NIC	Network Interface Controller

Table 1-4. Processor Terminology (Sheet 4 of 5)

Term	Description
OOB	Out of Band
Overshoot	The maximum voltage observed for a signal at the device pad, measured with respect to V_{CC}.
Pad	The electrical contact point of a semiconductor die to the package substrate. A pad is only observable in simulations.
PCI, PCIe*	Peripheral Component Interconnect (Express). In this document, this interconnect refers to the PCI logical layer used by the IOSF protocol.
PCI Express*	PCI Express Generation 2
PCODE	Power Management Unit Micro-code
PCS	Physical Coding Sub layer
PEC	Packet Error Code
PECI	Platform Environmental Control Interface
PHY	Physical Layer Device
Pin	The contact point of a component package to the traces on a substrate, such as the motherboard. Signal quality and timings are measured at the pin.
PLC	Platform LAN Connect
PMA	Physical Medium Attachment
PMC	Power Management Controller
PMD	Physical Medium Dependent
PMU	Power Management Unit
Power-Good	Power-Good, or PWRGOOD (an active high signal) indicates that all the system power supplies and clocks are stable. PWRGOOD does go active a predetermined time after system voltages are stable and does go inactive as soon as any of these voltages fail their specifications.
Processor	The 64-bit, single-core or multi-core component (package)
Processor Core	The term Processor Core refers to the silicon die itself which contains multiple execution cores. Each execution core has an instruction cache, data cache, and shares its 1-MB L2 cache with a sibling execution core.
PXE	Preboot Execution Environment
Rank	A unit of DRAM corresponding four to eight devices in parallel, ignoring ECC. These devices are usually, but not always, mounted on a single side of a DDR3 DIMM.
Ringback	The voltage to which a signal changes after reaching its maximum absolute value. Ringback is caused by reflections, driver oscillations, or other transmission line phenomena.
RMII	Reduced Media Independent Interface (Reduced MII)
RP	Root Port
RTC	Real Time Clock
SA	Source Address
SATA	Serial ATA
SDP	Software Defined Pins
SerDes	Serializer/Deserializer. A transceiver that converts parallel data to serial data and vice-versa.
SFD	Start Frame Delimiter
SGMII	Serialized Gigabit Media Independent Interface
SMBus	System Management Bus. A two-wire interface through which simple system and power management related devices communicate with the rest of the system. A bus carrying various manageability components, including the LAN controller, BIOS, sensors, and remote-control devices. SMBus is based on the operational principles of the I^2C* two-wire serial bus from Philips Semiconductor*. SMBus supports Alert signals for GbE and SMB_0 ports.
So-DIMM	Small outline Dual In-line Memory Module

Table 1-4. Processor Terminology (Sheet 5 of 5)

Term	Description
SPD	Serial Presence Detect
SPI	Serial Peripheral Interface
SSE	Intel® Streaming SIMD Extensions (Intel® SSE)
SSO	Simultaneous Switching Output (SSO) effects are differences in electrical timing parameters and degradation in signal quality caused by multiple signal outputs simultaneously switching voltage levels in the opposite direction from a single signal or in the same direction. These are called odd mode and even mode switching, respectively. This simultaneous switching of multiple outputs creates higher current swings that may cause additional propagation delay (“push-out”) or a decrease in propagation delay (“pull-in”). These SSO effects may impact the setup and/or hold times and are not always taken into account by simulations. System timing budgets include margin for SSO effects.
STR	Suspend To RAM
Stub	The branch from the bus trunk terminating at the pad of an agent.
SVR	Switched Voltage Regulator
TAC	Thermal Averaging Constant
TCO	Total Cost of Ownership (TCO) System Management
TDP	Thermal Design Power
TDR	Time Domain Reflectometry
TLP	Transaction Layer Packet in the PCI Express specification
Trunk	The main connection, excluding interconnect branches, from one end-agent pad to the other end-agent pad.
TSO	Transmit Segmentation Offload. A mode in which a large TCP/UDP I/O is handed to the device which the device then segments into L2 packets according to the requested MSS.
TSOD	Temperature Sensor On DIMM
UDIMM	Unbuffered Dual In-line Memory Module
UHCI	Universal Host Controller Interface
Undershoot	The minimum voltage extending below V_{SS} observed for a signal at the device pad.
Unit Interval (UI)	Binary signaling convention that is measure of time representing the transmission of a single data bit in a serial data stream. One bit is sent for every forwarded clock edge, whether it be a rising edge or a falling edge. If a number of edges are collected at instances t1, t2, tn,...., tk, then the UI at instance “n” is defined as the following: UI n = t n - t n-1.
USB	Universal Serial Bus
VCC	Processor core-power supply
Vcc_core	The core-power rail for the processor.
Victim	A network that receives a coupled crosstalk signal from another network is called the victim network.
VLAN	Virtual LAN
VPD	Vital Product Data (PCI protocol)
VRD	The Voltage Regulator Down (a down on the board solution) for the processor. The VRD is a DC-DC converter module that supplies the required voltage and current to a single processor.
VSS	Processor ground
x1	Refers to a link or port with one physical lane.
x4	Refers to a link or port with four physical lanes.
x8	Refers to a link or port with eight physical lanes.

1.7 Related Documents

Refer to the following documents for additional information.

Table 1-5. Processor Documents

Document	Document Number/ Location
VR12/IMVP7 Pulse Width Modulation (PWM) Specification – Product Specification	397113
Intel® Atom™ Processor C2000 Product Family - Platform Design Guide (PDG)	509126
Intel® Atom™ Processor C2000 Product Family for Microserver External Design Specification (EDS), Volume 1, 2, 3, and 4	508084
Intel® Atom™ Processor C2000 Product Family DDR3 UDIMM/SODIMM Trace Length Calculator (TLC)	507205
Intel® Atom™ Processor C2000 Product Family - RCC Platform Design Guide Supplement	511898
Intel® Atom™ Processor C2000 Product Family for Microserver Thermal and Mechanical Specifications and Design Guidelines	508956
Intel® Atom™ Processor C2000 Product Family BIOS Writer's Guide (BWG)	Vol. 1: 516815 Vol. 2: 516816
Manufacturing With Intel® FCBGA10/FCBGA11 Package Technologies	473867

Note:
1. For release dates or latest revisions and documentation numbers, contact the appropriate Intel field representative.

Table 1-6. Public Specifications (Sheet 1 of 2)

Document	Document Number/ Location
Address Resolution Protocol (ARP) Specification (RFC 826)	http://tools.ietf.org/html/rfc826
Advanced Configuration and Power Interface Specification, Revision 2.0b, October 2002	http://www.acpi.info
Advanced Configuration and Power Interface Specification 3.0	http://www.acpi.info
Alert Standard Format (ASF) Specification, Version 2.0	http://www.dmtf.org/standards/documents/ASF/DSP0136.pdf
AT Attachment – 6 with Packet Interface (ATA/ATAPI - 6)	http://T13.org (T13 1410D)
Crystal* Technical Glossary	Fox Electronics
Crystal Frequently Asked Questions	Fox Electronics
DDR3 SDRAM Specification	http://www.jedec.org
DMTF Network Controller Sideband Interface (NC-SI) Specification, Revision 1.0.0, May 2009	http://www.dmtf.org
Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision *1.0* (EHCI)	http://developer.intel.com/technology/usb/ehcispec.htm
EUI-64 Specification	http://standards.ieee.org/regauth/oui/tutorials/EUI64.html
Front Panel I/O *Connectivity Design Guide* (Look for I/O Connectivity in formfactors.org)	http://www.formfactors.org/developer/specs/A2928604-005.pdf
Institute of Electrical and Electronic Engineers (IEEE) Standard 802.3, 2008 Edition (Ethernet) Incorporates various IEEE standards previously published separately	http://standards.ieee.org
IEEE Standard 802.3, 2000 Edition	http://standards.ieee.org
IEEE Standard 1149.1, 2001 Edition (JTAG)	http://standards.ieee.org
IEEE Standard 1149.6, 2003 IEEE Standard for Boundary-Scan Testing of Advanced Digital Networks	http://standards.ieee.org
IEEE Standard 802.1Q for VLAN	http://standards.ieee.org
IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*, November 8, 2002	http://standards.ieee.org
IEEE 802.1AS Timing and Synchronization for Time Sensitive Applications in Bridged Local Area Networks Draft 2.0, February 22, 2008	http://standards.ieee.org
Intel® Packaging Information, Packaging Databook, 1999	http://www.intel.com/design/packtech/packbook.htm
Intel® 64 and IA-32 Architectures Software Developer's Manuals: • *Volume 1: Basic Architecture* • *Volume 2A: Instruction Set Reference, A-M* • *Volume 2B: Instruction Set Reference, N-Z* • *Volume 3A: System Programming Guide* • *Volume 3B: System Programming Guide* *Intel® 64 and IA-32 Architectures Optimization Reference Manual*	http://www.intel.com/products/processor/manuals/index.htm
IPv4 Specification (RFC 791)	
IPv6 Specification (RFC 2460)	
Low Pin Count Interface Specification, Revision 1.1 (LPC)	http://developer.intel.com/design/chipsets/industry/lpc.htm
Neighbor Discovery for IPv6 (RFC 4861)	
Multicast Listener Discovery (MLD) for IPv6 (RFC 2710)	
Multicast Listener Discovery Version 2 (MLDv2) for IPv6 (RFC 3810)	

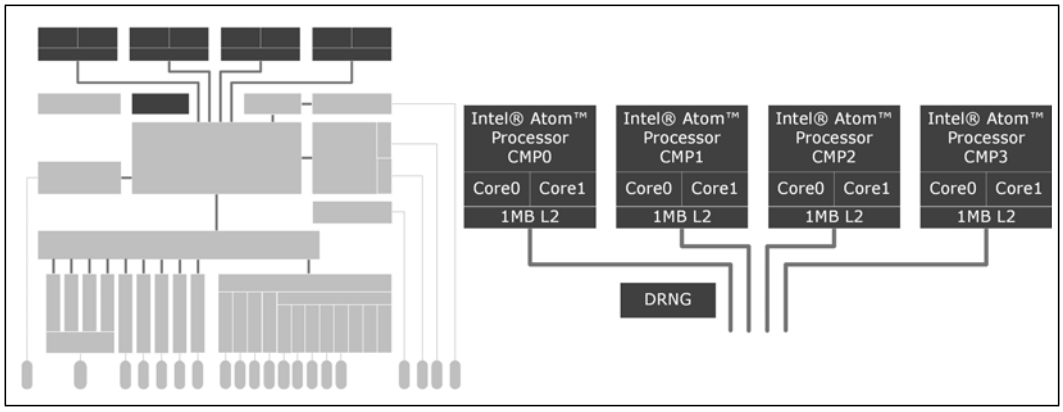
Table 1-6. Public Specifications (Sheet 2 of 2)

Document	Document Number/ Location
PCI Express Base Specification* - Revision 1.1	http://www.pcisig.com
PCI Express Base Specification - Revision 2.1	
PCIe Base Specification*, Revision 3.0	http://www.pcisig.com/specifications/pciexpress/base
PCIe Card Electromechanical Specification, Revision 1.1	http://www.pcisig.com/specifications/pciexpress/base
PCI Local Bus Specification 3.0	http://www.pcisig.com/specifications
PCI Power Management Specification, Revision 1.2	http://www.pcisig.com/specifications/conventional/pci_bus_power_management_interface/
PICMG3.1 Ethernet/Fiber Channel Over PICMG 3.0 Draft Specification. September 4, 2002, Version 0.90	http://www.picmg.org/newinitiative.stm
Quartz Crystal Theory of Operation and Design Notes	Fox Electronics: http://www.foxonline.com
Resonator Terminology and Formulas	Piezo Technology, Inc. http://www.piezotech.com
Serial ATA Specification, Revision 1.0a	http://www.serialata.org/specifications.asp
Serial ATA II: Extension to Serial ATA 1.0, Revision 1.0a	http://www.serialata.org/specifications.asp
Serial ATA II Cables and Connectors Volume 2 Gold	http://www.serialata.org/specifications.asp
Serial-GMII Specification Cisco Systems* Document ENG-46158, Revision 1.8	http://www.cisco.com/
System Management Bus (SMBus) Specification, SBS Implementers Forum, Version 2.0, August 2000	http://www.smbus.org/specs/
Transmission Control Protocol (TCP) Specification (Internet Standard RFC 793)	http://tools.ietf.org/html/rfc793
User Datagram Protocol (UDP) (Internet Standard RFC 768)	http://tools.ietf.org/html/rfc768
Universal Serial Bus Revision *2.0 Specification* (USB)	http://www.usb.org (or) http://www.usb.org/developers/estoreinfo/

§ §

2 Multi-Core Intel® Atom™ Processors

Figure 2-1. Multi-Core Block Diagram



2.1 Signal Descriptions

This portion of the SoC has no external signal pins.

2.2 Features

The C2xx0 is a highly-integrated System-on-a-Chip (SoC). The SoC is developed using the Intel® 22nm process and is based on the recently-enhanced Intel® Atom™ processor core.

The main architectural features are:

- Up to 8 Cores (with SKUs containing two, four, or eight cores)
  - — Xeon® series ISA compatibility
  - — VT-x2
  - — 1 MB shared L2 Cache (per 2 cores), 4 MB L2 total
  - — Turbo Support

2.3 SoC Components

2.3.1 SoC Core

The SoC core is designed using 22nm process technology for use in ultra-low-power applications. Each SoC module, called a CMP, incorporates dual CPU cores, a bus interface unit (BIU) and a 1 MB L2 cache.

The features of each SoC Module are:

- 2-wide out-of-order (OOO) scheduler
- Chip-Level Multi Processing (CMP). No support for hyper-threading technology.
- One thread per core
- Improved instruction fetch and decode functions
- Better branch predictors
- Improvements to TLB and caching hierarchy
- Hybrid OOO scheduling and OOO cache miss processing
- Per core-power gating support
- Intel® Streaming SIMD Extensions 4.1 and 4.2 (SSE4.1 and SSE4.2).
- Intel® 64 architecture
- Support for IA-32 instruction set
- Support for Intel® Virtualization Technology (Intel® VT) for IA-32, Intel® 64 and Intel® Architecture (Intel® VT-x)
- Support for Intel® Advanced Encryption Standard New Instructions (AES-NI)
- Support for a Digital Random Number Generator (DRNG)
- Intel® Turbo Boost Technology
- Supported C-states: C0, C1, C6C. The C4 state is not supported.

2.4 Features

2.4.1 Intel® Virtualization Technology

Intel® Virtualization Technology (Intel® VT) makes a single system appear as multiple independent systems to the software. This allows multiple, independent operating systems to run simultaneously on a single system. Intel VT comprises technology components to support virtualization of platforms based on Intel® architecture microprocessors and chipsets. Intel® Virtualization Technology for IA-32, Intel® 64 and Intel® Architecture (Intel® VT-x) added hardware support in the processor to improve the virtualization performance and robustness.

Intel VT-x specifications and functional descriptions are included in the *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3B* and is available at:

http://www.intel.com/products/processor/manuals/index.htm

Other VT documents are referenced at:

http://www.intel.com/technology/virtualization/index.htm

2.4.2 Intel® VT-x Objectives

- Robust - Virtual Machine Monitors (VMMs) no longer need to use paravirtualization or binary translation. This means that they run off-the-shelf operating systems and applications without any special steps.
- Enhanced - Intel VT enables VMMs to run 64-bit guest operating systems on IA x86 processors.
- More Reliable - Due to the hardware support, VMMs are now smaller, less complex, and more efficient. This improves reliability and availability and reduces the potential for software conflicts.
- More Secure - The use of hardware transitions in the VMM strengthens the isolation of VMs and further prevents corruption of one VM from affecting others on the same system. Intel VT-x provides hardware acceleration for virtualization of IA platforms. VMMs use Intel VT-x features to provide improved reliable virtualized platform.

2.4.2.1 Intel® VT-x Features

- Extended Page Tables (EPT)
  - An EPT is hardware-assisted page table physical memory virtualization.
  - The feature supports guest VM execution in unpaged protected mode or in real-address mode.
  - EPT eliminates VM exits from guest OS to the VMM for shadow page-table maintenance.
- Virtual Processor IDs (VPID)
  - The ability to use an assigned VM Virtual Processor ID to tag processor core hardware structures (such as TLBs) allowing a logic processor to cache information (such as TLBs) for multiple-linear address spaces.
  - This avoids flushes on VM transitions to give a lower-cost VM transition time and an overall reduction in virtualization overhead.
- Guest Preemption Timer
  - A mechanism for a VMM to preempt the execution of a guest OS VM after an amount of time specified by the VMM. The VMM sets a timer value before entering a guest.
  - The feature aids VMM developers in flexibility and Quality of Service (QoS) guarantees flexibility in guest VM scheduling and building QoS schemes.
- Descriptor-Table Exiting
  - Descriptor-table exiting allows a VMM to protect a guest OS from internal (malicious software based) attack by preventing relocation of key system data structures like Interrupt Descriptor Table (IDT), Global Descriptor Table (GDT), Local Descriptor Table (LDT), and Task Segment Selector (TSS).
  - A VMM using this feature intercepts (by a VM exit) attempts to relocate these data structures and prevent them from being tampered by malicious software.
- VM Functions
  - A VM function is an operation provided by the processor that is invoked using the VMFUNC instruction from guest VM without a VM exit.
  - A VM function to perform EPTP switching is supported and allows guest VM to load a new value for the EPT pointer, thereby establishing a different EPT paging structure hierarchy.

2.4.3 Security and Cryptography Technologies

2.4.3.1 Advanced Encryption Standard New Instructions (AES-NI)

The processor supports Advanced Encryption Standard New Instructions (AES-NI) that are a set of Single Instruction Multiple Data (SIMD) instructions that enable fast and secure data encryption and decryption based on the Advanced Encryption Standard (AES). AES-NI are valuable for a wide range of cryptographic applications, for example: applications that perform bulk encryption/decryption, authentication, random number generation, and authenticated encryption. AES is broadly accepted as the standard for both government and industry applications, and is widely deployed in various protocols. AES-NI consists of six Intel® SSE instructions. Four instructions, namely AESENC, AESENCLAST, AESDEC, and AESDELAST facilitate high performance AES encryption and decryption. The other two, AESIMC and AESKEYGENASSIST, support the AES key expansion procedure. Together, these instructions provide a full hardware for AES support, offering security, high performance, and a great deal of flexibility.

2.4.3.2 PCLMULQDQ Instruction

The processor supports the carry-less multiplication instruction, PCLMULQDQ. PCLMULQDQ is a Single Instruction Multiple Data (SIMD) instruction that computes the 128-bit carry-less multiplication of two, 64-bit operands without generating, and propagating carries. Carry-less multiplication is an essential processing component of several cryptographic systems and standards. Accelerating carry-less multiplication significantly contributes to achieving high-speed secure computing and communication.

2.4.3.3 Digital Random Number Generator

The processor introduces a software visible digital random number generation mechanism supported by a high-quality entropy source. This capability is available to programmers through the new RDRAND instruction. The resultant random number generation capability complies with existing industry standards (ANSI X9.82 and NIST SP 800-90). The instruction is described as RDRAND—Read Random Number in Volume 2 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*.

Some uses of the new RDRAND instruction include cryptographic key generation as used in a variety of applications including communication, digital signatures, secure storage, etc.

2.4.4 Intel® Turbo Boost Technology

Note: Intel Turbo Boost Technology may not be available on all SKUs.

Intel Turbo Boost Technology increases the ratio of application power to TDP. Thus, thermal solutions and platform cooling that are designed to less than thermal design guidance experience thermal and performance issues since more applications tend to run at the maximum power limit for significant periods of time. Refer to the *BIOS Writer's Guide (BWG) and the Turbo Implementation Guide* for more information.

- Intel Turbo Boost Technology is a feature that allows the processor to opportunistically and automatically run faster than its rated operating core and/or render clock frequency when there is sufficient power headroom, and the product is within specified temperature and current limits. The Intel Turbo Boost Technology feature increases performance of both multi-threaded and single-threaded workloads. The processor supports a turbo mode where the processor uses the thermal capacity associated with the package and run at power levels higher than TDP power for short durations. This improves the system responsiveness for short, bursty usage conditions. The turbo feature needs to be properly enabled by the BIOS for the processor to operate with maximum performance. Refer to the *BIOS Writer's Guide* for enabling details. Since the turbo feature is configurable and dependent on many platform design limits outside of the processor control, the maximum performance cannot be ensured. Turbo mode availability is independent of the number of active cores; however, the turbo mode frequency is dynamic and dependent on the instantaneous application power load, the number of active cores, user configurable settings, operating environment, and system design.

2.4.4.1 Intel® Turbo Boost Technology Frequency

The processor's rated frequency assumes that all execution cores are active and are at the sustained Thermal Design Power (TDP). However, under a typical operation not all cores are active or executing a high-power workload. Most applications are consuming less than the TDP at the rated frequency. Intel Turbo Boost Technology takes advantage of the available TDP headroom and active cores are able to increase their operating frequency. To determine the highest performance frequency amongst active cores, the processor takes the following into consideration to recalculate turbo frequency during runtime:

- Number of cores operating in the C0 state
- Estimated core current consumption
- Estimated package prior and present power consumption
- Package temperature

Any of these factors affect the maximum frequency for a given workload. If the power, current, or thermal limit is reached, the processor automatically reduces the frequency to stay with its TDP limit. Turbo processor frequencies are only active if the operating system is requesting the P0 state.

Core Turbo Boost frequencies may vary slightly from specified frequencies due to variances in the base clock frequency which is based on the installed DIMM speeds. The base clock frequency for 1600 MTS DIMMs will be 100 MHz while the base clock frequency for 1333 MTS DIMMs will be 83.3 MHz. This different base clock frequency and the fact that the Turbo Boost frequencies are derived based on a multiple of the base clock frequency may result in a slightly higher or slightly lower Turbo Boost Frequency than specified.

Table 2-1. Intel® Turbo Boost Core Frequency Overview

SoC	TDP Frequency	Number of Cores Active (i.e., not in C6)							
		1	2	3	4	5	6	7	8
C2750	2.4	2.6	2.6	2.6	2.6	2.6	2.6	2.6	2.6
C2730	1.7	2.4	2.4	2.3	2.3	2.2	2.2	2	2
C2550	2.4	2.6	2.6	2.6	2.6	x	x	x	x
C2530	1.7	2.4	2.4	2	2	x	x	x	x
C2350	1.7	2	2	x	x	x	x	x	x

2.5 CPUID Instruction and SoC Identification

The CPUID Instruction returns the processor identification and feature information in the EAX, EBX, ECX, and EDX registers. The instruction is described as CPUID—CPU Identification in Volume 2 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*.

For additional guidance, Intel offers Application Note 485 - *Intel Processor Identification and the CPUID Instruction*, Document No. 241618. This document is available through the local Intel sales representative.

Leaf 1 (EAX =1) of the CPUID returns the processor model, family, and stepping IDs as well as the processor features supported. Table 2-2, Table 2-3, and Table 2-4 show these fields for the SoC B0 and earlier steppings.

Table 2-2. CPUID Leaf 1 Instruction - EAX and EBX Registers

Bit Field	Value	Description	Notes
EAX[31:0:] - Identity			
31:28	0	*Reserved*	Family 6, Model 4Dh Intel® Atom™ processor family using the 22nm process
27:20	0	Extended Family ID	
19:16	4	Extended Model ID	
15:14	0	*Reserved*	
13:12	0	Processor Type ID	
11:8	6	Family ID	
7:4	Dh	Model ID	
3:0	0	Processor Stepping ID	
EBX[31:0:] - Identity and Features			
31:24	Assigned	Local APIC ID	ID assigned to the Local APIC for each processor thread during power-up.
23:16	10h	Maximum number logical processor IDs in SoC	The maximum range of APIC IDs that can be assigned. All SKUs have the same value.
15:8	08h	CLFLUSH instruction cache line size	In 8-byte increments: 8 x 8 = 64 bytes
7:0	0	Brand ID feature supported	Not supported

Table 2-3. CPUID Leaf 1 Instruction - ECX Register

Bit Field	Value	Feature Description
31	0	Zero (0)
30	1	RDRAND - On-chip Random Number Generator
29	0	F16C Support
28	0	AVX - Advanced Vector Extensions
27	0	OSXSAVE
26	0	XSAVE
25	1	AES Instruction Set
24	1	TSC - Deadline
23	1	POPCNT Instruction
22	1	MOVBE Instruction
21	0	x2APIC Support
20	1	SSE4_2 - SSE4.2 Instructions
19	1	SSE4_1 - SSE4.1 Instructions
18	0	DCA - Direct Cache Access
17	0	PCID - Process-Context Identifiers
16	0	Reserved
15	1	PDCM - Perfmon and Debug Capability MSR
14	1	xTPR Update Control
13	1	CMPXCHG16B Instruction
12	0	FMA - Fused Multiply-Add
11	0	Reserved
10	0	CNXT-ID - L1 Context ID
9	1	SSSE3 - Supplemental SSE3 Extensions
8	1	TM2 - Thermal Monitor 2
7	1	EST - Enhanced Intel SpeedStep® Technology
6	0	SMX - Safer Mode Extensions
5	1	VMX - Virtual Machine eXtensions
4	1	DS-CPL - CPL Qualified Debug Store
3	1	MONITOR - MONITOR/MWAIT Instructions
2	1	DTES64 - 64-Bit DS Area
1	1	PCLMULQDQ - Carry-Less Multiplication
0	1	SSE3 - SSE3 Extensions

Table 2-4. CPUID Leaf 1 Instruction - EDX Register

Bit Field	Value	Feature Description
31	1	PBE - Pending Break Enable Wake-up Support
30	0	Reserved
29	1	TM - Thermal Monitor
28	1	HTT - Multi-threading. If 1, the SoC can support more than one logical processor per package.
27	1	SS - Self Snoop Support
26	1	SSE2 - SSE2 Instruction Extensions
25	1	SSE - SSE Instruction Extensions
24	1	FXSR - FXSAVE, FXRSTOR Instructions
23	1	MMX - MMX Technology
22	1	ACPI - Thermal Monitor and Clock Control
21	1	DS - Debug Store
20	0	Reserved
19	1	CLFSH - CFLUSH Instruction
18	0	PSN - Processor Serial Number
17	1	PSE-36 - 36-Bit Page-Size Extension
16	1	PAT - Page Attribute Table
15	1	CMOV - Conditional Move/Compare Instruction
14	1	MCA - Machine Check Architecture
13	1	PGE - PTE Global Bit
12	1	MTRR - Memory Type Range Registers
11	1	SEP - SYSENTER and SYSEXIT Instructions
10	0	Reserved
9	1	APIC - APIC on Chip
8	1	CX8 - CMPXCHG8B Instruction
7	1	MCE - Machine Check Exception
6	1	PAE - Physical Address Extensions
5	1	MSR - RDMSR and WRMSR Support
4	1	TSC - Time Stamp Counter
3	1	PSE - Page Size Extensions
2	1	DE - Debugging Extensions
1	1	VME - Virtual-8086 Mode Enhancement
0	1	FPU - x87 FPU on Chip

The BIOS is able to determine the silicon stepping of the entire SoC. This is accomplished by reading the 32-bit CUNIT_MANUFACTURING_ID register in configuration space, bus 0, device 0, function 0, offset F8h. The SoC stepping is shown in the 8-bit field, MANUFACTURING_ID_BIT_7_0. Table 2-5 shows the information received when this register is read.

Table 2-5. SoC Stepping Information

Parameter	A0 SoC	A1 SoC	B0 SoC
Process	0.1	0.1	0.1
Manufacturing ID	0Fh	0Fh	0Fh
Manufacturing Stepping	0	1	2

In addition to verifying the processor signature, the BIOS needs the platform ID to properly target the microcode update. The platform ID is determined by reading bits [52:50] of the IA32_PLATFORM_ID register, (MSR 17h). This is a 64-bit register and is read using the RDMSR instruction. The three platform ID bits, when read as a Binary Coded Decimal (BCD) number, indicate the bit position in the microcode update header processor flags field that is associated with the installed processor.

§ §

Volume 2: Functional

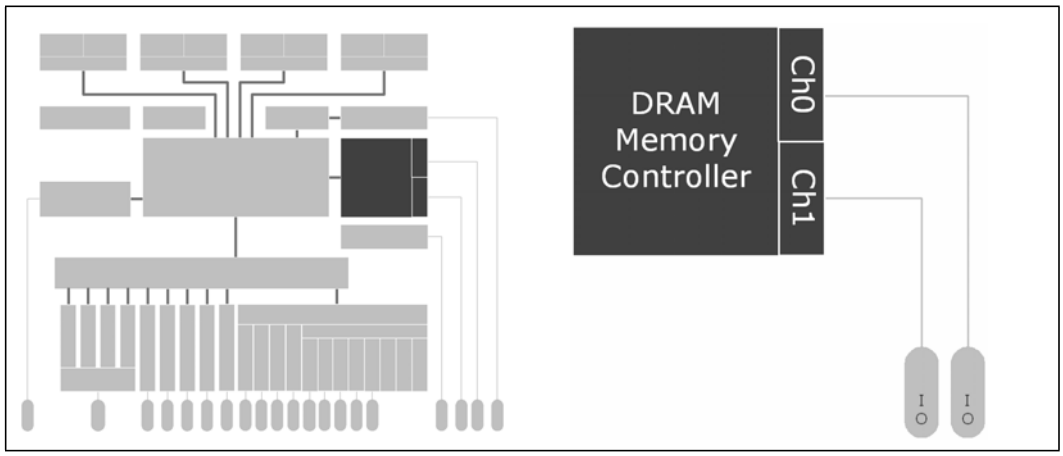
January 2016
Order Number: 330061-003US

3 Memory Controller

3.1 Introduction

The SoC Memory Controller supports up to 64 GB of native DDR3 (1.5V) and DDR3L (1.35V) by two independent memory controllers. The maximum capacity supported for each product SKU is shown in Table 1-2, "Intel® Atom™ Processor C2000 Product Family for Microserver Product SKUs" on page 35. Each controller supports up to two SODIMMs or UDIMMs per channel with a maximum date rate of 1600 MT/s. The memory controller supports a 64-bit data bus with 8-bit ECC and supports data transfer rates of 1333 MT/s and 1600 MT/s. The controller also supports non-ECC DDR3 DIMM memory. The supported DRAM chip data width is x8 and supported DRAM chip densities of 1 Gb, 2 Gb, 4 Gb and 8 Gb.

Figure 3-1. Memory Controller Covered in This Chapter



3.2 Signal Descriptions

The DDR3 signal details are provided in Chapter 31, "Signal Names and Descriptions."

3.3 Features

3.3.1 Supported Memory Configuration

The DDR3 memory controller contains two independent DDR3 memory controllers. Each memory channel supports either one or two DIMMs, where each DIMM is either single- or dual-rank. The supported DRAM chip data width is x8. The SoC does not support x16 devices.

When only one of the two memory channels is used in a platform board design, Channel 0 must be used. In all designs, Channel 0 must be populated by DRAM devices.

Within each memory channel DIMMs are populated in slot order; slot 0 is populated first and slot 1 last.

If a DIMM has two ranks, the ranks must be symmetrical (same chip width, same chip density, and same total memory size per rank).

If both memory channels of the memory controller are used, then both channels must be populated identically (same width, same density, same rank, and same total memory size per rank).

Minimum memory capacity:

- All products = 1 GB

Maximum memory capacity:

- C2750 and C2550 products = 64 GB (two memory controllers, 2 ranks each, 8 banks per rank, 8-Gbit-density components)
- C2730 and C2530 products = 32 GB (two memory controllers, 2 ranks each, 8 banks per rank, 4-Gbit-density components)
- C2350 product = 16 GB (one memory controller, 2 ranks, 8 banks per rank, 4-Gbit-density components)

3.3.2 System Memory Technology Supported

The SoC memory controller supports the following features:

- DDR3 (1.5V) and DDR3L (1.35V)
- ECC enabled DIMMs and SODIMMs
- Non-ECC UDIMMs
- 1600 or 1333 MT/s depending on SKU
- UDIMM, SODIMM, VLP DIMM, and memory (solder) down are supported
- Device width support for only x8 devices
- Device density: 1, 2, 4, or 8 Gb
- Number of ranks per channel: 1, 2, or 4
- DDR3 data scrambling to improve signal integrity (configurable)

3.3.3 System Memory Technology which is Not Supported

The SoC memory controller DOES NOT support the following features:

- DDR3 DRAM which supports Quad Rank technology is not supported
- DDR3 DRAM which supports Dual Die Stacking Technology is not supported

Table 3-1. Supported DDR3 Devices

Rank Size	DRAM Density	Data Width	Banks	Chips/Rank
1 GB	2 Gb	x8	8	8
2 GB	2 Gb	x8	8	8
4 GB	4 Gb	x8	8	8
8 GB	8 Gb	x8	8	8

Table 3-2. Supported DDR3 Memory Configurations

DIMM Size	# of Ranks Enabled	Rank Size	DRAM Chip Density	DRAM Chip Data Width	DRAM Chips/ DIMM	ECC Support
1 GB	1	1 GB	1 Gb	x8	9	Yes
2 GB	1	2 GB	2 Gb	x8	9	Yes
4 GB	2	2 GB	2 Gb	x8	18	Yes
4 GB	1	4 GB	4 Gb	x8	9	Yes
8 GB	2	4 GB	4 Gb	x8	18	Yes
8 GB	1	8 GB	8 Gb	x8	9	Yes
16 GB[1]	2	8 GB	8 Gb	x8	18	Yes

Note:
1. Pending DRAM technology availability.

Table 3-3. Supported DDR3 DRAM Timings

DRAM Speed Grade	DRAM Clock Frequency	Data Rate	Peak Bandwidth	Supported CL-tRCD-tRP
DDR3-1333	667 MHz	1333 MT/s	10.7 GB/s	9-9-9, 10-10-10
DDR3-1600	800 MHz	1600 MT/s	12.8 GB/s	11-11-11

Table 3-4. Supported Rank Population Configurations

DIMM 0		DIMM 1	
Rank0	Rank1	Rank2	Rank3
Enabled	Disabled	Disabled	Disabled
Enabled	Enabled	Disabled	Disabled
Enabled	Disabled	Enabled	Disabled
Enabled	Enabled	Enabled	Enabled

3.4 RAS Features

3.4.1 Data Parity Protection

The write and read data to and from the integrated memory controller (the internal unit in SSA handles I/O requests from the core and other bus agents, like SATA) are protected by even parity on each byte lane. Parity per byte lane is used since the internal data buffers contain byte write enables to support partial writes from the requesting agents.

3.4.2 Memory Controller Error Correcting Codes (ECC)

The DDR3 interface is protected by an ECC code for Single-Bit Error Correction (SEC) and Double-Bit Error Detection (DED). An 8-bit ECC code word is stored with every 8 bytes of data, which can protect a 128-bit wide interface. Since the data width is only 72 bits, 56 single-bit error syndrome codes are available.

In order for Advanced ECC Mode to operate optimally, setting the UCE_FILTER (Uncorrectable Error Fitler) register to “x11”, which will enable third level filtering, is recommended.

Note: An ECC DIMM only provides additional storage for redundant information, the actual error detection/correction takes place within the SoC memory controller.

One of these single-bit error syndrome codes converts from byte parity to the 8-byte ECC. When the write data ECC is generated, the parity is checked on all 8-byte lanes and the parity error signal generates the ECC. When checking ECC on a read, the check is performed assuming that the SoC D-Unit has no write data parity errors. Any uncorrectable error syndrome that is detected on a read results in the generation of bad parity for all 8-byte lanes.

When a word is written into ECC protected memory, the ECC bits are computed by a set of exclusive OR trees. When the word is read back, the exclusive OR trees use the data read from the memory to recompute the ECC. The recomputed ECC is compared to the ECC bits read from the memory. Any discrepancy indicates an error. By looking at which ECC bits do not match, identify which data or ECC bit is in error, or whether a double-bit error occurred. The result of this ECC calculation is called the syndrome. If the syndrome is zero, no error occurred. If the syndrome is non-zero, the syndrome is used to index Table 3-5 to determine which bits are in error, or if the error is uncorrectable.

Table 3-5 shows the ECC H-Matrix used by the memory controller. The table shows the supported single-error syndromes that are detectable during a read. The all zero syndrome indicates that no error was detected. All of the valid single error syndromes contain a odd number of 1s asserted. The 72 syndromes for data and ECC bits are treated as single-bit (correctable) errors. All other non-zero syndromes are treated as multiple-bit (uncorrectable) errors.

Table 3-5. Memory Controller ECC Syndrome Codes

ECC Bit	Syndrome
0	0000 0001
1	0000 0010
2	0000 0100
3	0000 1000
4	0001 0000
5	0010 0000
6	0100 0000
7	1000 0000

Parity Error	0110 0111

1-hot Syndrome Codes	
8	Possible
8	Used

3-hot Syndrome Codes	
56	Possible
56	Used

5-hot Syndrome Codes	
56	Possible
9	Used

7-hot Syndrome Codes	
8	Possible
0	Used

Data Bit	Syndrome
0	0010 0011
1	1000 1100
2	0001 1100
3	0110 0010
4	1100 0100
5	1010 0100
6	1101 0000
7	0100 0101
8	0010 1001
9	0000 1101
10	0001 1001
11	0100 1001
12	0100 1010
13	0011 1000
14	0001 0110
15	1001 0100
16	1010 0010
17	0101 1000
18	1001 1000
19	1110 0000
20	0010 1100
21	0000 1011
22	0100 0110
23	0100 1111
24	0001 0101
25	0011 0001
26	1111 0010
27	0010 0110
28	0000 1110
29	0101 0010
30	1000 0011
31	1111 0001

Data Bit	Syndrome
32	0100 0011
33	1010 0001
34	0000 0111
35	0111 0000
36	1011 0000
37	1000 1111
38	0110 1000
39	1100 0001
40	0010 1111
41	0010 1010
42	0101 0100
43	0001 0011
44	0011 0010
45	1100 0010
46	0010 0101
47	0110 0001
48	1111 0100
49	1010 1000
50	0001 1111
51	1000 0110
52	0001 1010
53	1100 1000
54	1001 0010
55	1111 1000
56	1001 0001
57	0101 0001
58	1000 0101
59	0110 0100
60	0011 0100
61	1000 1010
62	1000 1001
63	0100 1100

3.4.3 Demand and Patrol Scrubbing

Demand scrub is an operation when a read request encounters a correctable memory error and the read data is corrected (scrubbed) and written back to memory. Without demand scrub the corrected data is only delivered to the requester and the corrupted data is still left in memory. Demand scrub fixes the error in memory when it is detected, thus lowering the probability that a second error to the same 8B memory location would change the correctable error into an uncorrectable error.

When a correctable error is detected, the SoC integrated memory controller returns the corrected read data to the internal SoC buffer along with a correctable error notification. The buffer logic then marks the data buffer that receives the read data as dirty (modified), which causes an eventual writeback to memory. When the writeback occurs, the previously-corrected read data is written back to memory with the correct ECC, thus scrubbing the memory location before it can be read again. Demand scrub is only employed for fixing correctable ECC errors from a read to physical DRAM and should exclude MMIO access.

Patrol scrub is a method in which the cleanup process is initiated in the background by the internal SoC memory buffer. When patrol scrub is enabled, the buffer reads all of memory locations starting at Address 0, Rank 0 at a very low bandwidth for the purpose of fixing correctable errors. The patrol scrub agent issues read requests and does nothing with the read data (silently dropped). When a correctable error is detected, the memory controller performs the update and writes the modified data back to memory.

The BDPSCRUB (BDPSCRUB)—Offset 17Ah SoC sideband register is used to enable the patrol scrub engine. The BDPSCRUB register is also used to set the scrub period for the desired scrub rate. At the default scrub period, 8 GB of memory can be scrubbed in about 24 hours. The scrubbing process skips the low MMIO region and the upper bound is dictated by the amount of memory installed. Software is also provided the ability to set the start of the scrub address in the BDPSADDR (BDPSADDR)—Offset 17Bh SoC sideband register. Note that the scrub engine operates at the lowest priority level, which will not cause the memory to exit self-refresh.

Note: Additionally, the scrub engine does not ensure that the scrubs are issued at the specified rate; the specified rate is only the maximum rate. The scrub timer is ignored if the patrol scrub engine is waiting to issue a scrub request.

3.4.4 DDR3 Data Scrambling

Data scrambling is a technique to reduce supply noise and improve DRAM data signal integrity by XORing data bits and ECC bits in a pseudo random sequence. The pseudo random sequence has two important effects relative to power delivery. Across the 72 data/ECC bits of the bus, this feature ensures approximately 50% of the bits are logical 1 and the other 50% are logical 0 in every cycle. This eliminates the previous worst case where all bus bits simultaneously drive high or low. The second benefit of scrambling produces a white spectrum eliminating data dependent resonance patterns.

If these resonance patterns hit the correct frequency relative to the LC tank circuits in the power delivery network, they create significant amounts of supply noise. In terms of signal integrity, the worst case margin is empirically found when a large number of bits transitioning in a specific fashion to create the worst case ISI, crosstalk and supply noise simultaneously on a given victim bit. Data scrambling makes it unlikely that all these bits switch in the correct fashion, hitting these worst-case patterns.

Data scrambling is configurable during boot time by modifying register DSCRMBL[16]= 1 to enable and DSCRMBL[16]=0 to disable.

§ §

4 System Agent and Root Complex

4.1 Introduction

The C2xx0 contains a System Agent and Root Complex block that provides the main interface with the processor cores and the other SoC integrated elements. The I/O Fabric provides the connections to the PCI Express* Root Ports and the integrated I/O devices.

Figure 4-1. System Agent and Root Complex Covered in This Chapter

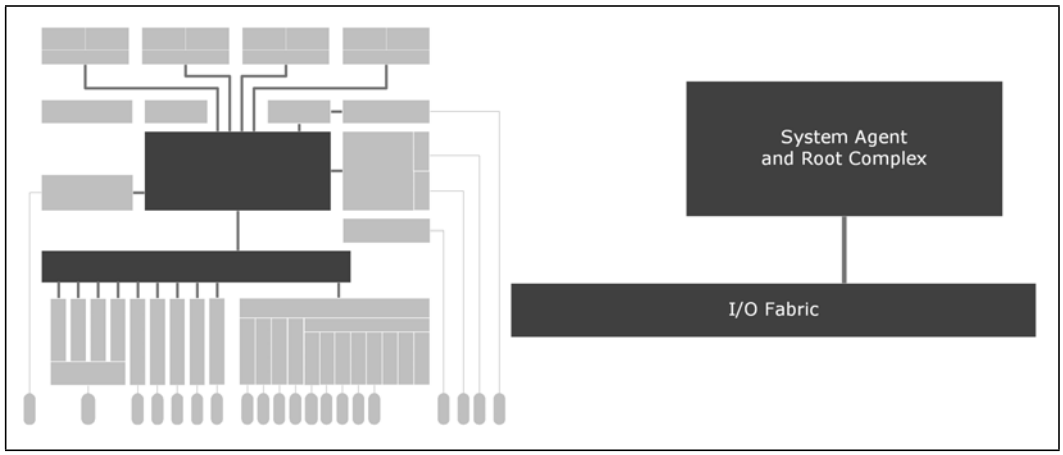


Table 4-1. References

Reference	Revision	Date	Document Title
PCI Express*	2.1	March 4, 2009	*PCI Express Base Specification*, Revision 2.1
Software Development Manual (SDM) www.intel.com Order Number: 325462	043	May 2012	*Intel® 64 and IA-32 Architectures Software Developer's Manual*

4.2 Signal Descriptions

While not shown precisely in the block diagram, five external signal pins are associated with this portion of the SoC. See Chapter 31, "Signal Names and Descriptions" for additional details.

The signal description table has the following headings:

- **Signal Name:** The signal/pin name
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

Table 4-2. Signals

Signal Name	Direction/ Type	Description
ERROR0_B ERROR1_B ERROR2_B	O	Error (active low) Detected errors are indicated to the external circuitry. • ERROR0_B indicates correctable errors. • ERROR1_B indicates non-fatal errors. • ERROR2_B indicates fatal errors. The platform board must ignore these SoC output signals while PMU_PLTRST_B (active-low SoC output) is asserted. *These signals are muxed and are used by other functions.*
MCERR_B	O	Machine Check Error (active low) Detected machine check errors (machine check exceptions) are indicated to the external circuitry. The platform board must ignore this SoC output signal while PMU_PLTRST_B (active-low SoC output) is asserted. *These signals are muxed and are used by other functions.*
IERR_B	O	Internal Error (active low) Detected unrecoverable internal errors are indicated to the external circuitry. The platform board must ignore this SoC output signal while PMU_PLTRST_B (active-low SoC output) is asserted. *These signals are muxed and are used by other functions.*

4.3 Features

- SoC System Agent (SSA) using the Pondicherry Intra-Die-Interconnect (IDI)
  - IDI is the standard interface between the caching agents of the core units and the SSA in the Pondicherry architecture.
- 36-bit physical memory-space addressing (64 GB)
- Patrol scrub engine that performs a memory scrub to fix correctable memory errors in the background
- Robust RAS
- Internal Root Complex Event Collector (RCEC) for PCI Express* and local error escalation
- PCI Express Advanced Error Reporting (AER) support
- MSI signaling
- INTx signaling
- Internal command and data path parity coverage (single bit each)
- Internal RAM parity coverage

4.4 Root Complex

The root complex, also called the Root Fabric (RTF), implements the bus 0 interconnect of a PCI Express* Root Complex. The root complex contains the internal Root Complex Event Collector (RCEC). Only one PCI hierarchy is in the SoC.

4.4.1 Transaction Flow

SoC internal transactions flow through the root complex fabric simultaneously in the upstream and downstream directions at full bandwidth. Peer-to-Peer (P2P) transaction routing is also not supported between the downstream root complex and I/O fabric ports. P2P memory read/writes are not supported.

4.4.2 Root Complex Primary Transaction Routing

The root complex provides primary transaction routing utilizing positive and negative decode. Positive decode routing is based on matching attributes between the internally-routed transaction and the attached root complex agents. The root complex has internal knowledge of its attached agent attributes to enable this decode.

Internal transaction routing is shown in Table 4-3. Transactions not listed in this table cause a negative decode. These negative-decode transaction include:

- Lock transaction
- I/O space transaction other than to a root port aperture
- Message transaction with a downstream destination
- Unsupported source-destination pair
- Transaction to disabled memory or I/O region
- Unrecognized transaction type
- Unrecognized address
- Unrecognized ID

Table 4-3. Root Complex Primary Transaction Routing

Transaction Type	Decode Type	Source	Destination
Memory Space	Base Address Register (BAR)	Any	PCIe* Root Ports
	Aperture in Memory Space	Any	PCIe Root Ports
	Refetchable Aperture	Any	PCIe Root Ports
	Message Signalled Interrupt (MSI)	Any	SoC System Agent
	HMBound Address	Any	SoC System Agent
	Address in the DOS Region	Any	SoC System Agent
I/O Space	Aperture in I/O Space	Any	PCIe Root Ports
Configuration Space	Type 0	SoC System Agent	PCIe Root Ports, Integrated devices
	Type 1 (Bridge)	SoC System Agent	PCIe Root Ports
Completion	ID Route	Any	PCIe Root Ports, SoC System Agent
Any	Negative Decode	Any	Integrated Devices *not the PCIe Root Port controllers* ***Note:*** Negative-decode transactions that have characteristics not capable of the integrated devices, such as an address width greater than 36 bits, or a payload of more than 64 bytes, are instead sent to the SoC Negative Decode Handler (NDH).

The SoC Negative Decode Handler (NDH) mentioned in Table 4-3 sends an Unsupported Request (UR) completion on reads and drops writes and completions. NDH event error logging occurs.

4.5 Reliability, Availability and Serviceability (RAS)

Reliability refers to how often errors occur in the system, and whether the system recovers from an error condition.

Availability refers to how flexible the system resources are allocated or redistributed for system utilizations and system recovery from errors.

Serviceability refers to how well the system reports and handles events related to errors.

The RAS features aim to achieve the following:

- Hardware-based error recovery on PCI Express* links.
  - — Packet re-transmission on detecting CRC errors.
- Clearly identify non-fatal errors and minimize fatal errors.
  - — Error reporting of the affected transactions by the appropriate completion responses or data poisoning.
  - — Correctable, non-fatal, and fatal errors are forwarded to the CPU via Non-Maskable Interrupt (NMI) or System Management Interrupt (SMI), or to an external device via the ERROR2_B, ERROR1_B, and ERROR0_B pins (see Table 32-6, “Core Misc Signals” on page 620).
  - — Error logging/reporting to assist error containment and recovery.

The Figure 4-2 shows the high-level SoC error handling scheme. The SoC receives the PCIe* error messages from downstream devices. The SoC logs these errors and other internal device errors.

4.6 Error Classification

Errors are classified as two types: uncorrectable and correctable. This classification separates those errors resulting in functional failures from those errors resulting in degraded performance. Uncorrectable errors are further classified as fatal or non-fatal. Classification of error severity as fatal, non-fatal, and correctable provides the platform with mechanisms for mapping the error to a suitable handling mechanism. Each severity triggers a system event according to the mapping defined by the System Event Map register. This mechanism provides the software the flexibility to map an uncorrectable error to the suitable error severity. For example, a platform may choose to map an uncorrectable ECC error as an non-fatal error while another platform design may require mapping the same error to a fatal error. The uncorrectable error mapping is set to the default mapping at power-on so it is consistent with the default mapping defined in Table 4-13. The software/firmware chooses to alter the default mapping after power-on.

4.6.1 Correctable Errors

Hardware correctable errors include those error conditions where the system recovers without any loss of information. The hardware corrects these errors, and no software intervention is required. For example, a link CRC error which is corrected by the data link level retry is considered a correctable error.

- An error is corrected by the hardware without software intervention. System operation is degraded, but its functionality is not compromised.
- A correctable error is logged and reported in a implementation specific manner:
  - — Upon the immediate detection of the correctable error, or
  - — Upon the accumulation of errors reaching to a threshold.

4.6.2 Fatal Errors

Fatal errors are uncorrectable error conditions which render the SoC hardware unreliable. For a fatal error, inband reporting to the CPU occurs. A reset is required to return to reliable operation.

- System integrity is compromised and continued operation is not feasible.
- System interface is compromised.
- Inband reporting is feasible, for example, an uncorrectable tag error in cache or a permanent PCIe* link failure.
- This error requires immediate logging and reporting of the error to the CPU.

4.6.3 Non-Fatal Errors

Non-fatal errors are software correctable or software/hardware uncorrectable errors which cause a particular transaction to be unreliable, but the system hardware is otherwise fully functional. Isolating non-fatal from fatal errors provides system management software the opportunity to recover from the error without reset and disturbing other transactions in progress. Devices not associated with the transaction in error are not impacted by the error. An example of recoverable error is an ECC uncorrectable error that affects only the data portion of a transaction.

- The error is not corrected by the hardware and requires software intervention for correction.
- Or the error is not corrected. Data integrity is compromised, but system operation is not compromised.
- This error requires immediate logging and reporting of the error to the CPU.

The OS/firmware takes the action to contain the error.

4.6.3.1 Software Correctable Errors

Software correctable errors are considered as a recoverable error. These errors include those error conditions where the system recovers without any loss of information. Software intervention is required to correct these errors.

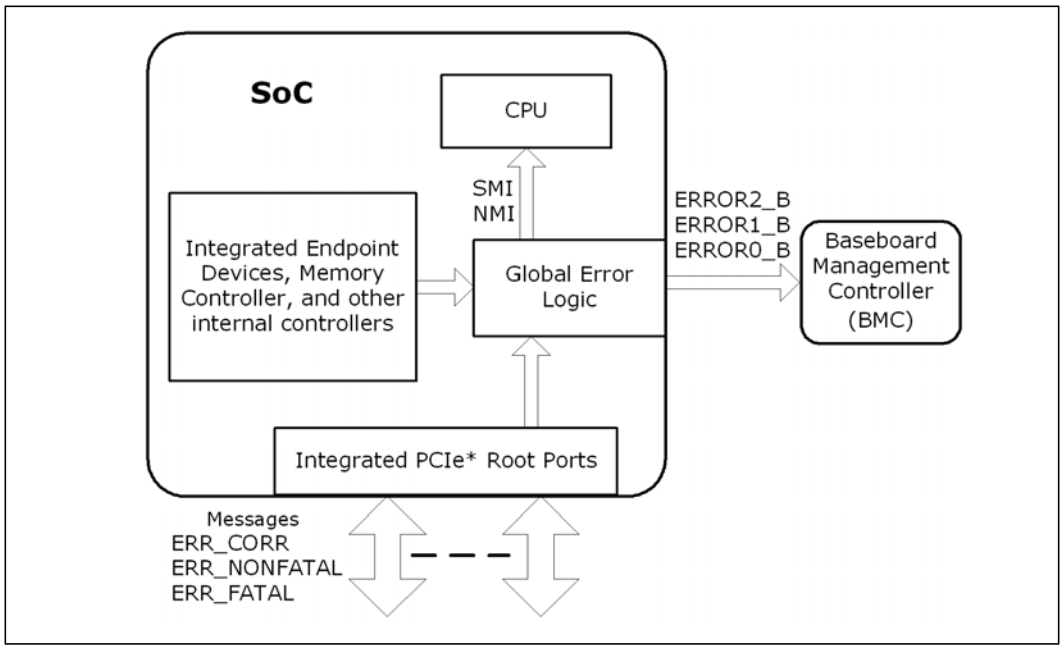
- This error requires immediate logging and reporting of the error to the CPU.
- The firmware or other system software layers take corrective actions.

Data integrity is not compromised with such errors.

4.7 Global Error Reporting

The SoC logs and reports the detected errors via system event generations. In the context of global error reporting, a system event is an event that notifies the system of the error. See Figure 4-2.

Figure 4-2. General Flow of SoC Error Reporting



PCI Express* error messages are received from downstream devices. The SoC logs these errors and other internal errors.

Two types of system events are generated:

- NMI or SMI to the CPU, or
- Error indication through the ERROR2_B, ERROR1_B, and ERROR0_B pins.

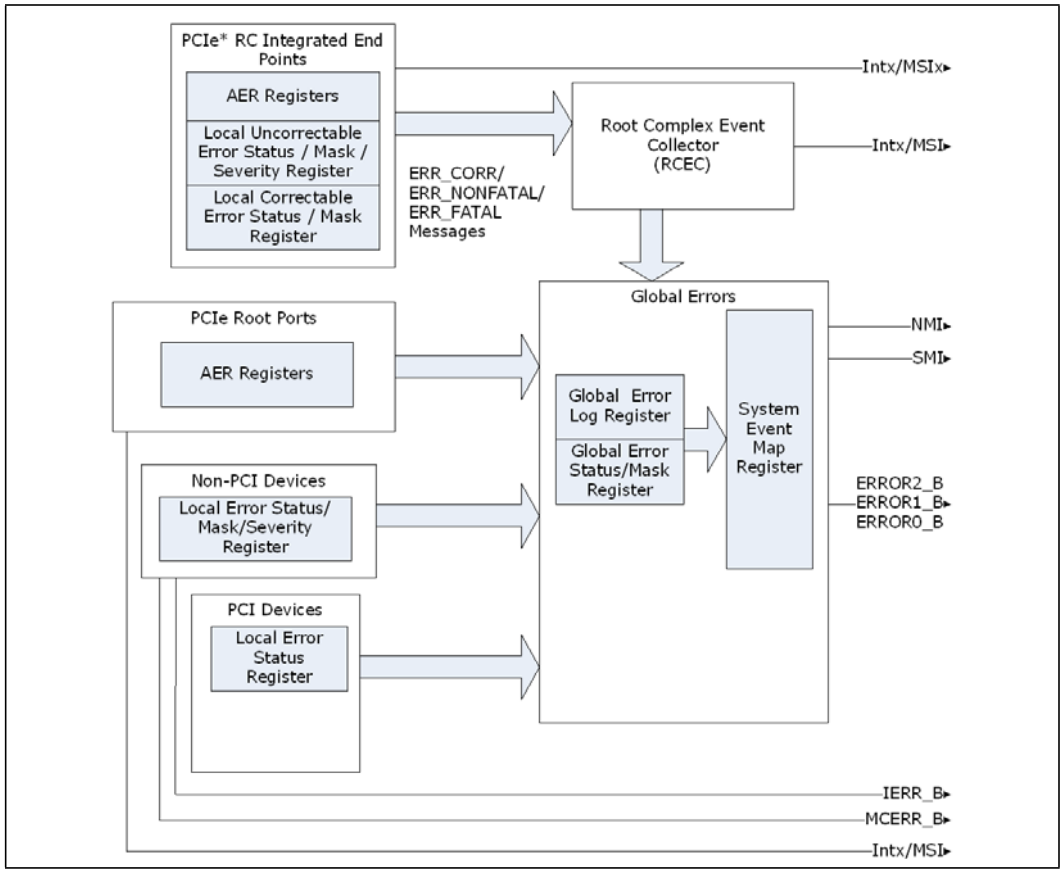
The CPU responds to a system event (NMI or SMI) and takes the appropriate action to handle the error.

An external agent such as a Baseboard Management Controller (BMC) monitors the active low, three error pins to determine the health of the SoC and interrupt the host CPU. In some severe error cases, when the CPU is no longer responding to system events resulting from an error(s), the three error pins provide a way to notify an external agent of the error. The external agent then performs a reset to recover the SoC functionality.

Machine check errors (machine check exceptions) and unrecoverable internal errors are also reported to the external circuitry through the SoC pins MCERR_B and IERR_B, respectively. Refer to Section 4.7.5, “MCERR/IERR Signaling” on page 89 for information about these pins.

A more detailed architectural view of all SoC error handling is shown in Figure 4-3.

Figure 4-3. Error Handling Architecture



The SoC detects errors from the PCIe* links and the SoC internal device errors. The errors are first logged and mapped to an error severity, and then mapped to a system event(s) for error reporting.

SoC error-reporting features are summarized below. Details are in the following sections.

- Detects and logs PCIe links and SoC internal device errors.
- First error/next error detection and logging for correctable/uncorrectable local errors.
- Allows flexible mapping of local uncorrectable errors to fatal or non-fatal error classes.
- First/next error detection and logging for correctable, non-fatal, and fatal global errors.
- Flexible error reporting using multiple reporting mechanisms.
- Supports PCIe error reporting mechanism based on the Root Complex Event Collector (RCEC).

The SoC provides direct mapping of system errors to NMI or SMI. The System Error (SERR) mechanism is not used to do this.

4.7.1 Reporting Errors to CPU

Detected errors are forwarded to the CPU using either an NMI or SMI.

4.7.1.1 Non-Maskable Interrupt (NMI)

Any error can be mapped to an NMI. However, NMIs are typically used to report fatal errors. When an error triggers, an NMI is generated to the CPU.

4.7.1.2 System Management Interrupt (SMI)

Any error can be mapped to an SMI. SMIs are typically used to report fatal, non-fatal, or correctable error conditions in the SoC. When an error triggers, an SMI is generated to the CPU.

4.7.2 Reporting Global Errors to an External Device

Detected errors are forwarded to an external device, a BMC for example, using the following active low, three error pins (see Section 32-6, "Core Misc Signals" on page 620):

- ERROR0_B - Correctable errors
- ERROR1_B - Non-fatal errors
- ERROR2_B - Fatal errors

4.7.3 Machine Check Architecture

This section provides the necessary details for the operating software to handle Machine Check Exceptions (MCE). Some operating systems hook the Machine Check Architecture (MCA) exception vector (18h) to allow system-crash analysis. Like some other Intel processors, the SoC has been enhanced to allow the machine state to be preserved across the assertion of the RESET# signal. The BIOS does not modify the MCA registers following the RESET# assertion. This allows the operating system to enhance the exception handler by having this information available following a reboot after an error has occurred. Only upon the assertion (the signal transition from low to high) of the COREPWROK input signal (indicating POWERGOOD, power-on) is the machine-check architecture state re-initialized.

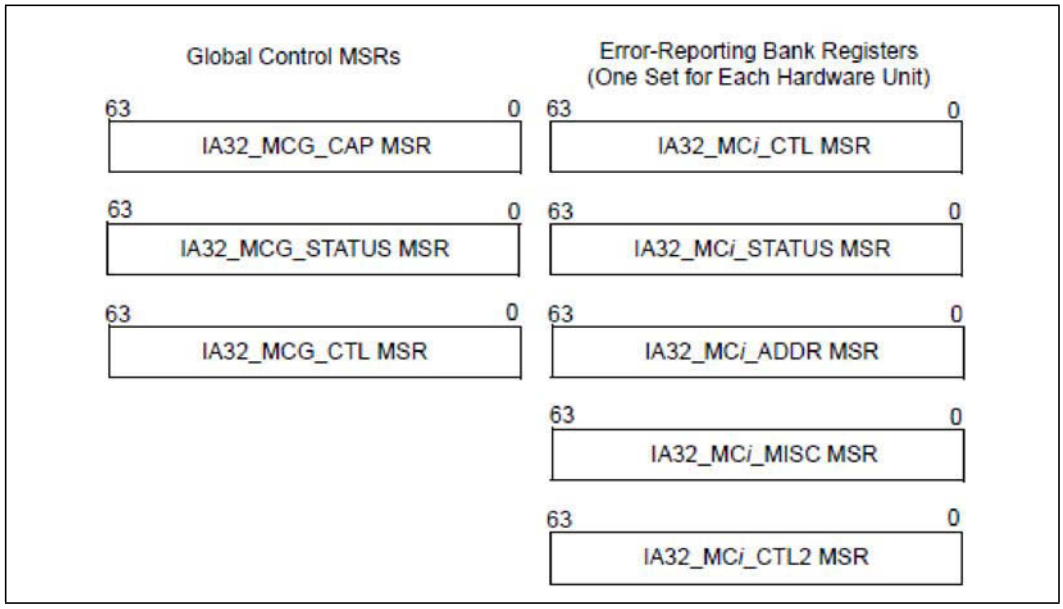
All MCA state information is accessible via the Model-Specific Register (MSR) accesses using the Read MSR (RDMSR) and Write MSR (WRMSR) instructions. RDMSR and WRMSR are described in Volume 2, Chapter 4 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*. See Section 1-6, "Public Specifications" on page 44.

There are three major classifications of MCA MSRs:

1. Global Control registers
2. Error-Reporting Bank registers
3. Extended Machine-Check State registers

The SoC does not have any MCA Extended Machine Check State registers. The first two classifications are shown in Figure 4-4.

Figure 4-4. Machine Check Global Control and Status Registers

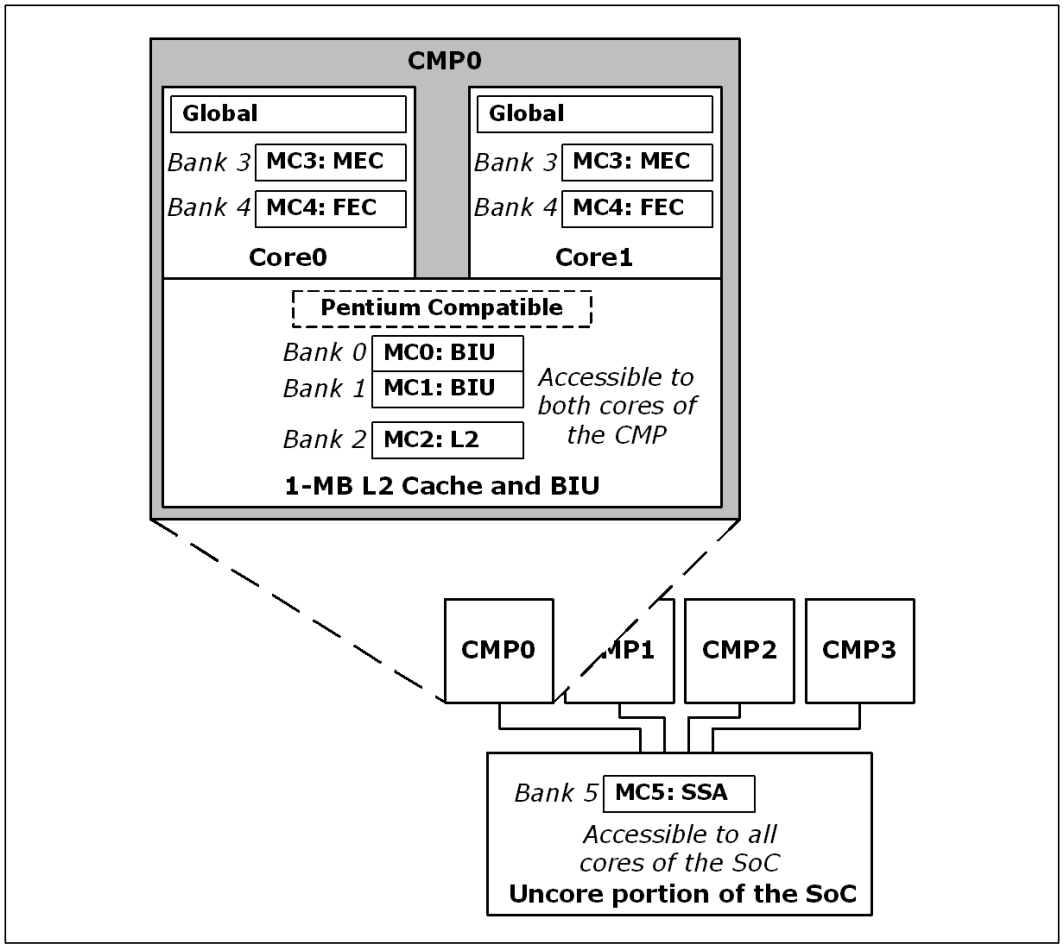


When a core executes the RDMSR or WRMSR instruction, some of the MSR information is located within the core itself. See Figure 4-5. The total number of cores in a particular SoC depends on the product SKU.

Other MSR information is located in the resources shared with the other core in the two-core module based on the Core Multi-Processor (CMP) technology. Each CMP contains two processor cores plus the L2 cache and Bus Interface Unit (BIU) they share.

The information for bank 5 is located in the uncore portion of the SoC and is accessible to all cores in the SoC.

Figure 4-5. Physical Locations of the MCA Register Information



Notes: In Figure 4-5:

1. Pentium® Compatibility:
   IA32_P5_MC_ADDR (MSR 0h)
   IA32_P5_MC_TYPE (MSR 1h)
2. Machine Check (MC) Global Model-Specific Registers (MSRs):
   IA32_MCG_CAP (MSR 179h)
   IA32_MCG_STATUS (MSR 17Ah)
3. Banks 0 and 1:
   BIU = Bus Interface Unit
4. Bank 2:
   L2 = Level-2 shared 1-MB Cache
5. Bank 3:
   MEC = Memory Execution Cluster
6. Bank 4:
   FEC = Front-End Cluster, includes the Instruction Cache
7. Bank 5:
   SSA = SoC System Agent

4.7.3.1 Machine Check Availability and Discovery

The Machine Check Architecture (MCA) and Machine Check Exception (MCE) are model-specific features. Software can execute the CPUID instruction (with EAX = 1) to determine that the particular core processor implements these features. Following the execution of the CPUID instruction (with EAX = 1), the SoC settings of the MCA feature bit 14, and MCE feature bit 7, are both set to 1 indicating that the features are available.

4.7.3.1.1 Machine Check Discovery Algorithm

1. Execute the CPUID instruction with EAX = 0. Check that the returned vendor ID equals Genuine Intel.
2. Execute the CPUID instruction with EAX = 1 to get the feature flags. Ensure that the MCA feature flag (bit 14) is set to a 1 and the MCE feature flags (bit 7) is set to a 1.
3. Read the Machine Check Capabilities register (IA32_MCG_CAP, MSR 179h) and get the bank count from IA32_MCG_CAP[7:0]. The value is 6 indicating that banks 0 through 5 exist for the SoC. Bank 0 MSRs begin at MSR 400h.
4. Read IA32_MCG_CAP[9]. This value is 0 in that extended state registers are not supported by the SoC.
5. Read IA32_MCG_CAP[8]. This value is 0 in that the IA32_MCG_CTL register (MSR 17Bh) is not supported by the SoC.
6. The BIOS writes all 1s to the six IA32_MCi_CTL registers. The “i” indicates banks 0 through 5.
7. Only if power-on RESET# occurred, the IA32_MCi_STATUS registers are cleared.
8. If power-on RESET# was not detected, then the BIOS may optionally log the reported errors as a CPU error or an other equivalent platform error in an event log.

4.7.3.2 P5 Compatibility MSRs

IA32_P5_MC_ADDR (MSR 0h)

IA32_P5_MC_TYPE (MSR 1h)

Newer software should not use the two P5 Compatibility MSRs. In the context of P5 the SoC does not return anything meaningful. Instead, the software needs to use IA32_MC0_STATUS (MSR 401h) instead of MSR 1h and IA32_MC0_ADDR (MSR 402h) instead of MSR 0h.

4.7.3.3 Machine Check Global Control MSRs

These machine check architecture MSRs and bit fields are defined in Section 15.3.1 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*. Brief SoC-specific descriptions follow.

4.7.3.3.1 Machine Check Global Capabilities Register (MSR 179h)

IA32_MCG_CAP

A read-only, 64-bit MSR determines the capabilities of the machine check architecture of the SoC:

- Bits [63:25] = Reserved.
- Bit 24 = MCG_SER_P - Software Error Recovery Support Present flag. This bit is 0 for the SoC indicating software error recovery is not supported, and MSR_MCG_CONTAIN (MSR 178h) is not available and should not be accessed.
- Bit [23:16] (8 bits) = MCG_EXT_CNT - Number of Extended Machine Check State registers. The SoC does not have any MCA Extended Machine Check State registers. See bit 9 of this register.
- Bits [15:12] = Reserved.
- Bit 11 = MCG_TES_P - Threshold-Based Error Status Present Flag. This bit is 1 for the SoC indicating that it provides threshold-based error status in bits [56:53] of the IA32_MC*i*_STATUS register. This error-status feature is also called the yellow/green health reporting.
- Bit 10 = MCG_CMCI_P - Corrected Machine Check (Interrupt) Error Counting/Signaling Extension Present Flag. This bit is 0 for the SoC indicating it does not support an extended state nor the associated MSRs necessary to support the reporting of an interrupt on a corrected error event and/or the threshold counting of corrected errors.
- Bit 9 = MCG_EXT_P - Extended MSRs Present flag. This bit is 0 for the SoC indicating that it does not have any MCA Extended Machine Check State registers which, if they existed, would start at MSR address 180h.
- Bit 8 = MCG_CTL_P - Control MSR Present flag. This bit is 0 for the SoC indicating the SoC does not have the IA32_MCG_CTL register defined as MSR 17Bh by MCA.
- Bit [7:0] (8 bits) = Count field. Indicates the number of hardware unit error-reporting banks available in a particular processor implementation. This field is 6 for the SoC indicating it has six error-reporting banks (0, 1, 2, 3, 4, and 5). In this document, the letter "i" in a register name represents the bank number of the MSR. The First Error-Reporting register, IA32_MC0_CTL (bank 0) always starts at MSR address 400h.

4.7.3.3.2 Machine Check Global Status Register (MSR 17Ah)

IA32_MCG_STATUS

A 64-bit, read-only MSR determines the current state of the machine check architecture of the SoC after an error has occurred. The SoC hardware sets and clears these bits. Only three bits are defined:

- Bits [63:3] = Reserved.
- Bit 2 = MCIP - Machine Check In Progress flag. This bit is set when the execution of the SoC machine check handler begins and can be cleared by the software. If another machine check exception is signaled while this bit is set, the machine enters the shutdown state.
- Bit 1 = EIPV - Error IP Valid flag. This flag is always 0 for the SoC indicating that the Instruction Pointer (IP) pushed onto the stack does not necessarily point to the instruction that caused the exception.
- Bit 0 = RIPV - Restart IP Valid flag. Restart is never possible with the SoC, and so this bit is always 0.

4.7.3.3.3 IA32_MCG_CTL Not Provided (MSR 17Bh)

The SoC does not provide this 64-bit MSR. The Control MSR Present (MCG_CTL_P) flag of the IA32_MCG_CAP (MSR 179h) is 0 indicating that the IA32_MCG_CTL register is not present.

4.7.3.4 Machine Check Error-Reporting MSR Banks 0-5

The SoC has six sets of machine check error-reporting MSR banks that reside at MSR addresses 400h through 416h. See Table 4-4. These 64-bit registers are described in this section.

Table 4-4. SoC MC Bank MSR Addresses

Portion of SoC	Hardware Unit	MCA Bank Number	IA32_MCi_CTL	IA32_MCi_STATUS	IA32_MCi_ADDR	IA32_MCi_MISC	IA32_MCi_CTL2
CMP	BIU	MC0	0x400	0x401	x402		
CMP	BIU	MC1[1]	0x404	0x405			
CMP	L2	MC2	0x408	0x409	0x40A		
Core	MEC	MC3	0x40C	0x40D	0x40E		
Core	FEC	MC4	0x410	0x411	0x412		
Uncore	SSA	MC5	0x414	0x415	0x416		

1. The MC1 bank is provided for compatibility with existing operating systems. While the name MC1 is mentioned here, the RMSR instructions to MC1 are ignored and the RMSR instruction never reports errors nor has any enable bits.

A shaded cell means that the MSR register is not implemented for the particular machine check bank. The physical portions of the SoC are shown in Figure 4-5.

4.7.3.4.1 General Description of Registers

IA32_MCi_CTL

In general, the IA32_MCi_CTL registers function in a similar fashion. Each is a 64-bit, read/write MSR. Each bit of the 64-bit register can be set by the software to enable or disable an individual error-reporting condition.

The SoC cores do not alias IA32_MC0_CTL to the EBL_CR_POWERON (MSR 2Ah).

The BIOS sets this register in all six banks to all ones (FFFF_FFFF_FFFF_FFFFh) even though some bits are unused and are not error-enable bits. Even so, when the software reads bits that correspond to unimplemented error conditions, a zero is always returned. The setting of this register does not affect the logging of errors but only the reporting of exceptions for uncorrectable errors. The errors are always logged.

IA32_MCi_STATUS

The IA32_MCi_STATUS registers contain information related to a machine check error if the VAL bit (bit 63 of the particular IA32_MCi_STATUS register) is set. These registers follow a general format shown in Volume 3, Chapter 15 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*. The software clears this MSR by explicitly writing 0 to it. Writing any other value causes a general protection exception. As mentioned previously, this register is not reset by the hardware and retains the prior values across warm resets. Refer to the IA32_MCi_STATUS descriptions for each bank in the following subsections.

IA32_MCi_ADDR

If implemented by the bank, the 64-bit IA32_MCi_ADDR register contains the address of the code or data memory location that produced the machine check error if the ADDRV flag (bit 58) in the IA32_MCi_STATUS register is set. This register must not be read if the corresponding ADDRV flag is not set. This register is updated according to the same rules regarding the overwriting of errors in the corresponding IA32_MCi_STATUS register. Writing anything but 0 to an implemented IA32_MCi_ADDR MSR causes a general protection exception. Like the IA32_MCi_STATUS register, IA32_MCi_ADDR is not reset by the hardware and retains the prior values across warm resets.

For MC1 (bank 1, BIU), a read or write to MSR 406h causes a general protection exception.

IA32_MCi_MISC and IA32_MCi_CTL2

These machine check MSRs are not implemented in the SoC. A read or write to these MSRs causes a general protection exception.

The six machine check error-reporting banks for the SoC are described in the following subsections.

4.7.3.4.2 Bank 0 — BIU IA32_MC0_CTL (MSR 400h)

Machine check error reporting for the Bus Interface Unit (BIU) associated with the particular core uses two machine check banks: MC0 and MC1. The MC0 bank is described first.

This document does not provide details of the individual Error-Reporting Enable flags of the 64-bit IA32_MC0_CTL register. Refer to the general description of "IA32_MCi_CTL."

4.7.3.4.3 Bank 0 — BIU IA32_MC0_STATUS (MSR 401h)

This is a 64-bit register with read/write, zero-to-clear, sticky access. The IA32_MC0_STATUS register follows the descriptions shown for IA32_MCi_STATUS in Volume 3, Chapter 15 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*:

- Bit 63 = VAL - IA32_MC0_STATUS register valid.
- Bit 62 = OVER - Error overflow.
- Bit 61 = UC - Uncorrected error. See Table 4-6.
- Bit 60 = EN - Error reporting enabled.
- Bit 59 = MISCV - IA32_MC0_MISC register valid. This is always 0.
- Bit 58 = ADDRV - IA32_MC0_ADDR register valid. See Table 4-6.
- Bit 57 = PCC - Processor context corrupted. See Table 4-6.
- Bits [56:53] = Reserved.
- Bits [52:38] = Other Information (IA32_MCG_CAP bit 10 is 0) - For the SoC, this bit field contains Intel-internal information. See the additional Other Information bit field below.
- Bits [37:32] = Other Information - For the SoC, this bit field contains Intel-internal information.
- Bits [31:16] = MSCOD Model-Specific Error Code - For the SoC, this bit field contains Intel-internal information.
- Bits [15:0] are the MCA Error Code for the BIU bank MC0. See Table 4-5. Refer to Section 15.9 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual* for information about interpreting the Simple and Compound MCA Error codes.

Table 4-5. IA32_MC0_STATUS

MCA Error Code (Bits [15:0])	Error-Code Encoding	Error Type	PCC (57)	ADDRV (58)	UC (61)
0x0003	Simple	External Error	0	0	0
0x0400	Simple	Internal Timer Error	1	0	1
0x0410	Simple	Internal Unclassified Error	1	Varies	1
0x0420	Simple	Internal Unclassified Error	1	0	1
0x0810	Compound	Bus/Interconnect Error	1	Varies	1
0x0820	Compound	Bus/Interconnect Error	1	Varies	1

4.7.3.4.4 Bank 0 — BIU IA32_MC0_ADDR (MSR 402h)

This is a 64-bit register with read/write, zero-to-clear, sticky access. It contains information that enables Intel to isolate the BIU bus-interface access that caused the last update to the IA32_MC0_STATUS register if the ADDRV bit of that register is set.

4.7.3.4.5 Bank 1 — BIU IA32_MC1_CTL (MSR 404h)

MC1 is shown in the SoC MSR list and can be read (RMSR) and written (WMSR) as an MCA bank; the WMSR instructions are ignored, and the RMSR instructions do not show enable bits or errors. The MC1 bank is provided only for compatibility with existing operating systems.

4.7.3.4.6 Bank 1 — BIU IA32_MC1_STATUS (MSR 405h)

MC1 is shown in the SoC MSR list and can be read (RMSR) and written (WMSR) as an MCA bank; the WMSR instructions are ignored, and the RMSR instructions do not show enable bits or errors. The MC1 bank is provided only for compatibility with existing operating systems.

4.7.3.4.7 Bank 2 — L2 IA32_MC2_CTL (MSR 408h)

Machine check error reporting for the L2 cache logic associated with the particular core uses the machine check bank MC2.

This document does not provide details of the individual Error-Reporting Enable flags of the 64-bit IA32_MC2_CTL register. Refer to the general description of “IA32_MCi_CTL.”

4.7.3.4.8 Bank 2 — L2 IA32_MC2_STATUS (MSR 409h)

This is a 64-bit register with read/write, zero-to-clear, sticky access. The IA32_MC2_STATUS register follows the descriptions shown for IA32_MCi_STATUS in Volume 3, Chapter 15 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*:

- Bit 63 = VAL - IA32_MC2_STATUS register valid.
- Bit 62 = OVER - Error overflow.
- Bit 61 = UC - Uncorrected error. See Table 4-6.
- Bit 60 = EN - Error reporting enabled.
- Bit 59 = MISCV - IA32_MC2_MISC register valid. This is always 0.
- Bit 58 = ADDRV - IA32_MC2_ADDR register valid. See Table 4-6.
- Bit 57 = PCC - Processor context corrupted. See Table 4-6.
- Bits [56:55] = Reserved.
- Bits [54:53] = Yellow/Green Tracking Bits.
- Bits [52:38] = Other Information (IA32_MCG_CAP bit 10 is 0) - See the additional Other Information bit definition below.
  When bits [15:0] = 0x080F (MCA Error Code = PIC Error), bits [52:38] = 0.
  - Bit 52 = Count Overflow of L2 cache lines with single-bit errors.
  - Bits [51:45] = Reserved.
  - Bits [44:38] = Count of L2 cache lines with single-bit errors.
- Bits [37:32] = Other Information - Bank specific.
  When bits [15:0] = 0x080F (MCA Error Code = PIC Error), bits [37:32] = 0.
  - Bits [37:36] = Reserved.
  - Bit 35 = Misc. L2 Tag error.
  - Bit 34 = Core ID.
  - Bits [33:32] = Array with error:
    00 - Data
    01 - Tag
    10 - State
    11 - Reserved
- Bits [31:16] = MSCOD Model-Specific Error Code - Bank specific. For the SoC, this bit field contains Intel-internal information.
- Bits [15:0] are the bank-specific MCA error code for the L2 bank MC2. Table 4-6 shows these error codes and the treatment of the PCC (bit 57), ADDRV (bit 58), and UC (bit 61) fields of IA32_MC2_STATUS.

Table 4-6. IA31_MC2_STATUS

MCA Error Code (Bits [15:0])	Error	Detecting Event	PCC (57)	ADDRV (58)	UC (61)
0x010A or 0x110A	Correctable Errors	Cache Read	0	1	0
0x010A	Uncorrectable Errors	Cache Read	1	1	1
0x080F	PIC Errors	Invalid PIC request	1	1	1

Note: The address captured (as indicated by ADDRV being set) is always the complete physical address of the cache access that discovered the error even if that physical address is not the precise address associated with the location that contains the error.

Note: The model-specific error codes allow for the identification of the array reporting the error in cases where the different arrays generate the same architecturally defined MCA error code.

4.7.3.4.9 Bank 2 — L2 IA32_MC2_ADDR (MSR 40Ah)

The IA32_MC2_ADDR registers contain the full-physical address of the cache access that caused the last update to the IA32_MC2_STATUS register.

4.7.3.4.10 Bank 3 — MEC IA32_MC3_CTL (MSR 40Ch)

Machine check error reporting for the Memory Execution Cluster (MEC) of the particular core uses the machine check bank MC3.

This document does not provide details of the individual Error-Reporting Enable flags of the 64-bit IA32_MC3_CTL register. Refer to the general description of “IA32_MCi_CTL.”

4.7.3.4.11 Bank 3 — MEC IA32_MC3_STATUS (MSR 40Dh)

This is a 64-bit register with read/write, zero-to-clear, sticky access. The IA32_MC3_STATUS register follows the descriptions shown for IA32_MCi_STATUS in Volume 3, Chapter 15 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*:

- Bit 63 = VAL - IA32_MC3_STATUS register valid.
- Bit 62 = OVER - Error overflow.
- Bit 61 = UC - Uncorrected error. See Table 4-7.
- Bit 60 = EN - Error reporting enabled.
- Bit 59 = MISCV - IA32_MC3_MISC register valid. This is always 0.
- Bit 58 = ADDRV - IA32_MC3_ADDR register valid. See Table 4-7.
- Bit 57 = PCC - Processor context corrupted. See Table 4-7.
- Bits [56:53] = Reserved.
- Bits [52:38] = Other Information when IA32_MCG_CAP [10] = 0 (default). Not Defined.
- Bits [37:32] = Other Information. Not Defined.
- Bits [31:16] = MSCOD Model-Specific Error Code - Bank specific. For the SoC, this bit field is described in Table 4-7.
- Bits [15:0] are the bank-specific MCA error code for the MEC. Table 4-7 shows these error codes and the treatment of the PCC (bit 57), ADDRV (bit 58), and UC (bit 61) fields of IA32_MC3_STATUS.

Table 4-7. IA32_MC3_STATUS

Model-Specific Error Code (Bits [31:16])	Error	MCA Error Code (Bits [15:0])	Detecting Event	PCC (57)	ADDRV (58)	UC (61)
0x0001	L1 Data Cache Correctable Parity Error	0x0135	Data Read	0	1	0
		0x0165	Prefetch			
		0x0175	Eviction			
		0x0185	Snoop			
0x0011	L1 Data Cache Uncorrectable Parity Error	0x0135	Data Read	1	1	1
		0x0165	Prefetch			
		0x0175	Eviction			
		0x0185	Snoop			
0x0012	L1 Data Tag Array Uncorrectable Parity Error	0x0135	Data Read	1	1	1
		0x0165	Prefetch			
		0x0175	Eviction			
		0x0185	Snoop			
		0x0151	Instruction Fetch			

Note: The address captured (as indicated by ADDRV being set) is always the complete physical address of the cache access that discovered the error even if that physical address is not the precise address associated with the location that contains the error (as might be the situation in the case of the L1 Data Tag Array Uncorrectable Parity Error).

Note: The model-specific error codes allow for the identification of the array reporting the error in cases where the different arrays generate the same architecturally defined MCA error code.

4.7.3.4.12 Bank 3 — MEC IA32_MC3_ADDR (MSR 40Eh)

The IA32_MC3_ADDR registers contain the full-physical address of the cache access that caused the last update to the IA32_MC3_STATUS register.

4.7.3.4.13 Bank 4 — FEC IA32_MC4_CTL (MSR 410h)

Machine check error reporting for the Fetch Execution Cluster (FEC) of the particular core uses the machine check bank MC4.

This document does not provide details of the individual Error-Reporting Enable flags of the 64-bit IA32_MC4_CTL register. Refer to the general description of "IA32_MCi_CTL."

4.7.3.4.14 Bank 4 — FEC IA32_MC4_STATUS (MSR 411h)

This is a 64-bit register with read/write, zero-to-clear, sticky access. The IA32_MC4_STATUS register follows the descriptions shown for IA32_MCi_STATUS in Volume 3, Chapter 15 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*:

- Bit 63 = VAL - IA32_MC4_STATUS register valid.
- Bit 62 = OVER - Error overflow.
- Bit 61 = UC - Uncorrected error. See Table 4-8.
- Bit 60 = EN - Error reporting enabled.
- Bit 59 = MISCV - IA32_MC4_MISC register valid. This is always 0.
- Bit 58 = ADDRV - IA32_MC4_ADDR register valid. See Table 4-8.
- Bit 57 = PCC - Processor context corrupted. See Table 4-8.
- Bits [56:53] = Reserved.
- Bits [52:38] = Other Information (IA32_MCG_CAP bit 10 is 0) - See the additional Other Information bit definition below.
  When bits [15:0] = 0x080F (MCA Error Code = PIC Error), bits [52:38] = 0.
  - Bit 52 = Count overflow of L2 cache lines with single-bit errors.
  - Bits [51:45] = Reserved.
  - Bits [44:38] = Count of L2 cache lines with single-bit errors.
- Bits [52:38] = Other Information when IA32_MCG_CAP [10] = 0 (default). Not Defined.
- Bits [37:32] = Other Information. Not Defined
- Bits [31:16] = MSCOD Model-Specific Error Code - Bank specific. For the SoC, this bit field is described in Table 4-8.

Bits [15:0] are the bank-specific MCA error code for the MEC. Table 4-8 shows these error codes and the treatment of the PCC (bit 57), ADDRV (bit 58), and UC (bit 61) fields of IA32_MC4_STATUS.

Table 4-8. IA32_MC4_STATUS

Model-Specific Error Code (Bits [31:16])	Error	MCA Error Code (Bits [15:0])	Detecting Event	PCC (57)	ADDRV (58)	UC (61)
0x0001	L1 Instruction Cache Correctable Parity Error	0x0151	Instruction Fetch	0	1	0
0x0002	L1 Instruction Tag Correctable Parity Error	0x0151	Instruction Fetch	0	1	0
		0x0181	Snoop			
0x0003	Internal Parity Error	0x0005	Unspecified	1	1	1

Note: The address captured (as indicated by ADDRV being set) is always the complete physical address of the cache access that discovered the error even if that physical address is not the precise address associated with the location that contains the error.

Note: The model-specific error codes allow for the identification of the array reporting the error in cases where the different arrays generate the same architecturally defined MCA error code.

4.7.3.4.15 Bank 4 — FEC IA32_MC4_ADDR (MSR 412h)

The IA32_MC4_ADDR registers contain the full-physical address of the cache access that caused the last update to the IA32_MC4_STATUS register.

4.7.3.4.16 Bank 5 — SSA IA32_MC5_CTL (MSR 414h)

Machine check error reporting for the SoC System Agent (SSA) uses the machine check bank MC5. The BIOS is able to affect the behavior of processing transactions with an uncorrectable error for the MC5 machine check bank. Refer to the MC_SIGNAL_MODE bit of the 32-bit sideband BMCMODE_LOW register located at sideband port 3, offset 5Ch.

This document does not provide details of the individual Error-Reporting Enable flags of the 64-bit IA32_MC5_CTL register. Refer to the general description of "IA32_MCi_CTL."

4.7.3.4.17 Bank 5 — SSA IA32_MC5_STATUS (MSR 415h)

This is a 64-bit register with read/write, zero-to-clear, sticky access. The IA32_MC5_STATUS register follows the descriptions shown for IA32_MCi_STATUS in Volume 3, Chapter 15 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*:

- Bit 63 = VAL - IA32_MC5_STATUS register valid.
- Bit 62 = OVER - Error overflow. Indicates a second error occurred while a previous error was still valid.
- Bit 61 = UC - Uncorrected error. Is 0 for corrected error.
- Bit 60 = EN - Error reporting enabled.
- Bit 59 = MISCV - IA32_MC5_MISC register valid. This is always 0.
- Bit 58 = ADDRV - IA32_MC5_ADDR register valid.
- Bit 57 = PCC - Processor context corrupted, uncorrected error.
- Bits [56:53] = Reserved.
- Bits [52:38] = Other Information (IA32_MCG_CAP bit 10 is 0) - See the additional Other Information bit definition below.
  - Bit 52 = Corrected Overflow - When set, indicates an overflow of the corrected error count.
  - Bits [51:38] = Corrected Error Count - Value indicates the number of corrected errors received.
- Bits [37:32] = Other Information - Bank specific.
  - Bit 37 = Internal Buffer RAM Error.
    1 - Indicates a parity error was detected on the internal buffer RAM.
    0 - Indicates a data error was received from a PFI- or IDI-requesting agent.
  - Bits [36:32] = Value represents the identification of a requesting agent that forwarded a data error to the SSA.Bits [31:16] = MSCOD Model-Specific Error Code - Bank specific. For the SoC, this bit field is described in Table 4-9.
- Bits [15:0] are the bank-specific MCA error code for the MEC. Table 4-9 shows these error codes and the treatment of the PCC (bit 57), ADDRV (bit 58), and UC (bit 61) fields of IA32_MC5_STATUS.

Table 4-9. IA32_MC5_STATUS

Error	Model-Specific Error Code (Bits [31:16])	MCA Error Code (Bits[15:0])	Detecting Event
Corrected Error or Uncorrected Error	0x0090	0x0090	Read to DDR3 Channel 0
	0x0091	0x0091	Read to DDR3 Channel 1
	0x00A0	0x00A0	Write to DDR3 Channel 0
	0x00A1	0x00A1	Write to DDR3 Channel 1
Uncorrected Error	0x009F	0x009F	Read to internal buffer RAM
	0x00AF	0x00AF	Write to internal buffer RAM

Note: The address captured (as indicated by ADDRV being set) is always the complete physical address of the cache access that discovered the error even if that physical address is not the precise address associated with the location that contains the error.

Note: The model-specific error codes allow for the identification of the array reporting the error in cases where the different arrays generate the same architecturally defined MCA error code.

4.7.3.4.18 Bank 5 — SSA IA32_MC5_ADDR (MSR 416h)

The IA32_MC5_ADDR registers contain the full-physical address of the DDR3 SDRAM or internal buffer RAM access that caused the last update to the IA32_MC5_STATUS register.

4.7.4 Error-Status Cloaking Feature

Error-status cloaking is an error management feature that allows the platform board management firmware to intercept corrected and uncorrected errors before the operating system software reads and clears the error log.

A new Model-Specific Register (MSR) has been added to enable or disable this feature. SMM_MCA_CONTROL is a 64-bit register at MSR 52h which contains 2 bits for this feature. This MSR is accessible only while the thread is executing in the System Management Mode (SMM). There is an SMM_MCA_CONTROL MSR register for each logical processor (has a thread scope). If accessed when not in SMM, a General-Protection Exception (#GP) is generated. See Table 4-10.

Table 4-10. SMM_MCA_CONTROL - MSR 52h - Enable/Disable Error-Status Cloaking Feature

Bits	Default	Name	RDMSR	WRMSR	Action
63:10	0	Reserved	0	#GP[1]	
9	0	PEND_SMI_ON_MCA	Allowed	Allowed	Post a pending SMI
8:1	0	Reserved	0	#GP[1]	
0	0	CERR_RD_STATUS_IN_SMM_ONLY	Allowed	Allowed	When 1, a valid corrected error status (V = 1, UC = 0, PCC = 0) is visible only when read in SMM. If not in SMM, a 0 status is returned for the valid corrected errors.

1. A General-Protection Exception (#GP) is generated.

4.7.4.1 Hide Corrected-Error Status From OS

Normally, corrected errors are reported through the IA32_MCi_STATUS registers which are accessible through the Read MSR (RDMSR) and Write MSR (WRMSR) instructions. Even though the SoC does not support Corrected Machine Check Interrupt (CMCI), these errors are logged in IA32_MCi_STATUS and are visible to the operating system. Here the IA32_MCi_STATUS contains the Valid bit = 1 (bit 63), the Uncorrected Error bit = 0 (bit 61), and the Processor Context Corrupted (PCC) bit = 0 (bit 57).

When the cloaking feature is enabled (SMM_MCA_CONTROL[0] =1), the operating system is prevented from reading these logged valid corrected errors unless the software is operating in the SMM. If not in SMM with the cloaking feature enabled, the RDMSR instruction can access IA32_MCi_STATUS, but the instruction returns 0 even though the valid corrected error(s) is logged.

This feature can be dynamically enabled/disabled.

4.7.4.2 SMI for MCA Uncorrected Errors

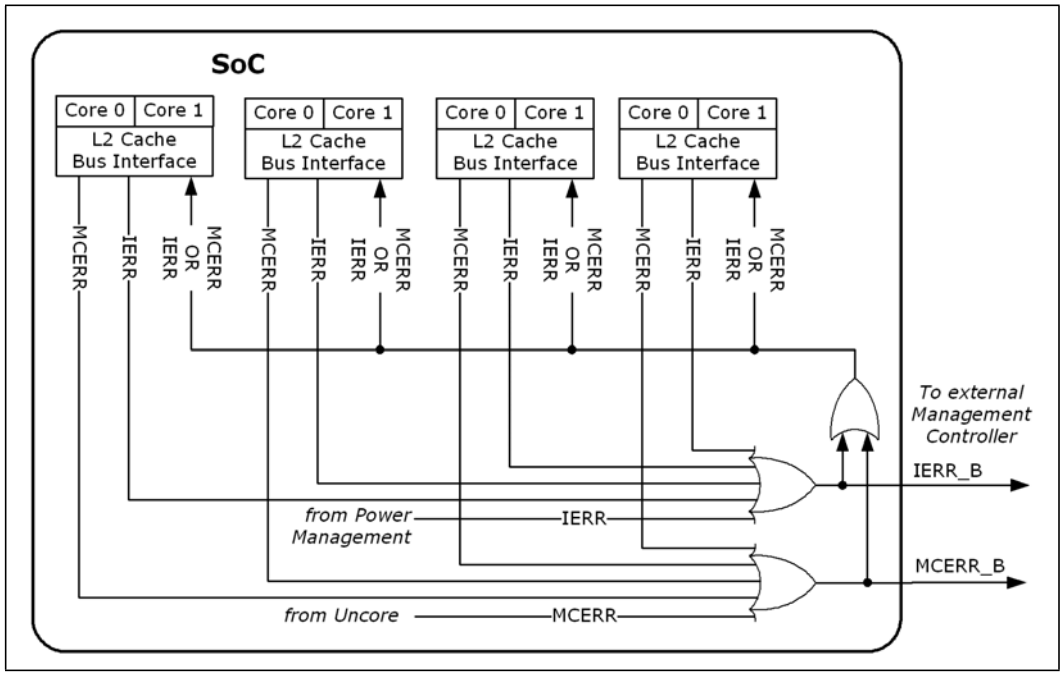
When the PEND_SMI_ON_MCA feature is enabled (SMM_MCA_CONTROL[9] =1), a System Management Interrupt (SMI) is made pending whenever the Machine Check Architecture (MCA) mechanism processes uncorrectable errors. Here the IA32_MCi_STATUS contains the Valid bit = 1 (bit 63), the Uncorrected Error bit = 1 (bit 61), and the Processor Context Corrupted (PCC) bit = 0 (bit 57).

This feature can be dynamically enabled/disabled.

4.7.5 MCERR/IERR Signaling

The SoC escalates machine check errors (MCERR_B) and unrecoverable internal errors (IERR_B) to the external BMC and to the CPU cores for handling non-maskable and other fatal errors in the system. See Figure 4-6. The MCERR and IERR signals are broadcast to the CPU and to the external platform.

Figure 4-6. MCERR and IERR Handling



The IERR signal signifies a catastrophic internal error, a condition which requires immediate attention or possibly shutdown. When this error occurs, the processor core may not be able to execute reliably through the INT18 handler. The following are some possible cases of such catastrophic errors:

- Retirement watchdog time-out from the core.
- Internal error detected by the SoC power management circuitry.

The MCERR signal signifies a machine check error occurred and that SoC Machine Check Architecture registers, accessible through the MSRRD and MSRWR instructions, may have additional information concerning the error.

Note: Board designs must not consider IERR_B and MCERR_B valid until after the PMU_PLTRST_B (Platform Reset) signal is deasserted by the SoC. When the SoC is powered-up or a cold boot, the IERR_B and MCERR_B signals may be unstable and falsely signal an internal error or machine check error before the platform reset is deasserted by the SoC.

4.7.6 PCI Express INTx and MSI

PCIe* INTx and MSI are supported through the PCIe standard error reporting. The SoC forwards the MSI generated from the downstream PCIe devices to the CPU. Also, PCIe Root Ports and the Root Complex Event Collector (RCEC) in the SoC generates INTx/MSI interrupts for error reporting if enabled. Refer to the *PCI Express Base Specification*, Revision 2.1 for more details on the PCIe standard and the Advanced Error Reporting (AER) capability.

4.7.7 Error Register Overview

The SoC contains a set of error registers to support error reporting. These error registers are assumed to be sticky unless specified otherwise. Sticky means the register values are retained even after a hard reset—they are only cleared by the software or by a power-on reset.

The two levels of hierarchy for the error registers are:

- Local Error registers
- Global Error registers

The Local Error registers are associated with the SoC local devices (GbE, SMBus, Root Complex, PCIe Root Ports, SoC system agent, memory controller, SATA2, SATA3, USB2 and platform controller unit). The Global Error registers collect the errors reported by the Local Error registers and map them to system events.

The four types of local devices are:

- Non-PCI devices
- Legacy PCI devices
- PCI Express devices
- PCI Express Root Ports

The non-PCI devices, the SoC memory controller as an example, directly report errors to the global error logic. These devices use a proprietary mechanism for reporting errors to the global error logic.

Also, the SoC system agent generates a MCERR when an internal parity error or a DDR3 ECC error is detected. MCERR is also generated when an internal unexpected completion is detected. Furthermore, the internal power management unit generates an IERR when errors are detected.

The legacy PCI devices (SATA2, SATA3, USB2 and the platform controller unit) have limited error-logging capabilities. These devices support PCI registers and report errors to the global error logic.

The PCIe root complex integrated endpoints (GbE, SMBus, and Root Complex) implement the PCIe Advanced Error Reporting (AER) capability and report errors to the global error logic through the Root Complex Event Collector (RCEC). These PCIe integrated endpoint devices support AER registers for logging and reporting internal-fabric and device-specific errors. Device-specific errors are logged in the Local Error registers and reported.

These PCIe integrated endpoint devices generate the PCIe error messages ERR_CORR, ERR_NONFATAL, and ERR_FATAL to the RCEC. These are errors that originate from the Root Complex.

The RCEC also supports the PCIe AER capability and generates INTx/MSI interrupts per the PCI Express Base Specification, Revision 2.1. The errors reported to the RCEC optionally signal to the SoC global error logic according to their severities through the programming of the PCIe Root Control register (ROOTCTL). Messages are generated, logged, forwarded, and ultimately notified to the global error logic.

The PCIe Root Ports support the PCIe AER capability and generate INTx/MSI interrupts per the PCI Express Base Specification, Revision 2.1. Also, the PCIe Root Ports optionally signal to the SoC global error logic according to their severities through the programming of the PCIe Root Control register (ROOTCTL). When the system error reporting is enabled for the specific PCIe error type, the SoC maps the PCIe error to the SoC error severity and reports the error to the Global Error Status register.

4.7.7.1 Local Error Registers

Each local device contains a set of local error registers. The PCIe Root Port Local Error registers are defined by the *PCI Express Base Specification*, Revision 2.1.

The local error register definitions are:

- Local Uncorrectable Error Status Register

  The SoC provides the Local Uncorrectable Error Status registers for the uncorrectable errors associated with the SoC local interfaces. When a specific uncorrectable error occurred in the local interface, its corresponding bit in the Uncorrectable Error Status register is set. Each error is individually masked by the Uncorrectable Error Mask register.

- Local Uncorrectable Error Mask Register

  The SoC provides the Local Uncorrectable Error Mask registers for the uncorrectable errors associated with the SoC local interfaces. Each error detected by the Local Uncorrectable Error Status register is individually masked by the Uncorrectable Error Mask register. If an error is masked, the corresponding status bit is not set for any subsequent detected error. A masked error (respective bit set in the Mask register) is not recorded or reported in the Uncorrectable Header Log register and does not update the uncorrectable First Error Register (FERR)/Next Error Register (NERR).

- Local Uncorrectable Error Severity Register

  The SoC provides Local Error Severity registers for uncorrectable errors associated with SoC local interfaces. The Local Uncorrectable Error Severity register controls whether an individual error is reported as a non-fatal or fatal error. An error is reported as fatal when the corresponding error bit in the severity register is set. If the bit is clear, the corresponding error is considered non-fatal.

Note: The PCIe Root Complex integrated endpoint detected uncorrectable internal errors are reported using Uncorrectable Internal Error Status (bit 22) of the ERRUNCSTS register and then uses the ERRUNCSEV: Uncorrectable Internal Error Severity bit to control the severity of the uncorrectable internal errors. This is reported using Uncorrectable Internal Error Status (bit 22) of the ERRUNCSTS register and the Corrected Internal Status (bit 14) of the ERRCORSTS register.

- Local Uncorrectable First/Next Error Status Register

  The SoC provides the Local Error Log register for the errors associated with the SoC local interfaces. When an error is detected by the SoC, the information related to the error is stored in the log register. The SoC local errors are first separated into correctable and uncorrectable categories. Each category contains two sets of log registers: FERR and NERR. FERR logs the first occurrence of an error, while NERR logs the subsequent occurrence of the errors.

Note: FERR/NERR do not log a masked error. The FERR log remains valid and unchanged from the first error detection until the clearing of the corresponding FERR Error bit in the Error Status register by the software. The xxxxERRUNCSTS registers are only cleared by writing to the corresponding Local Error Status register. For example, clearing bit 0 in RTF_ERRUNCSTS clears the bit in this register and bit 0 in RTF_FERRUNCSTS and RTF_NERRUNCSTS.

- Local Uncorrectable First Error Header Log Register

  The SoC provides Local First Error Header Log register for the uncorrectable errors associated with the SoC local interfaces. The Header log stores the header information of the associated first uncorrectable error.

Note: Only the Root Complex, the SMBus, and the D-Unit local devices have xxxxFERRUNCHDRLOG registers.

- Local Correctable Error Status Register

  The SoC provides the Local Correctable Error Status registers for the correctable errors associated with the SoC local interfaces. When a specific correctable error occurs in the SoC local interface, its corresponding bit in the Correctable Error Status register is set. Each error is individually masked by the Correctable Error Mask register.

Note: Only the GbE and the memory controller local devices have correctable errors.

- Local Correctable Error Mask Register

  The SoC provides the Local Correctable Error Mask registers for the correctable errors associated with the SoC local interfaces. Each error detected by the Local Correctable Error Status register is individually masked by the Correctable Error Mask register. If an error is masked, the corresponding status bit is not set for any subsequent detected error. A masked error (respective bit set in the Mask register) is not recorded or reported in the Correctable Header Log register and does not update the correctable FERR/NERR registers.

- Local Correctable First/Next Error Status Register

  The SoC provides the Local Correctable Error Log register for the errors associated with the SoC local interfaces. When a correctable error is detected by the SoC, the information related to the first/next error is stored in the xxxxFERRCORSTS/ **NERRCORSTS registers. FERR logs the first occurrence of an error, while NERR logs the subsequent occurrence of the errors.

Note: FERR/NERR do not log a masked error. The FERR log remains valid and unchanged from the first error detection until the clearing of the corresponding FERR Error bit in the Error Status register by the software. The xxxxERRCORSTS registers are only cleared by writing to the corresponding Local Error Status register. For example, clearing bit 0 in DUNIT_ERRCORSTS clears the bit in this register and bit 0 in DUNIT_FERRCORSTS and DUNIT_NERRCORSTS.

- Local Correctable First Error Header Log Register

  The SoC provides the Local First Error Header Log register for the correctable errors associated with the SoC local interfaces. The Header log stores the header information of the associated first correctable error.

- Local Uncorrectable MCERR/IERR Register

  The SoC P-Unit provides these registers for recording uncorrectable Machine Check Error (MCERR) and Internal Error (IERR) status.

4.7.7.2 Global Error Registers

The global error registers collect the errors reported by the local interface and convert the error to system events.

- Global Error Mask/Status Register

  The SoC provides three global error status registers to collect the errors reported by the SoC clusters—Global Fatal Error Status (GFERRSTS), Global Non-Fatal Error (GNERRSTS) Status, and Global Correctable Error Status (GCORERRSTS). Each register has an identical format that each bit in the register represents the fatal, non-fatal, or correctable error reported by its associated interface: memory controller, SoC system agent, PCIe Root Ports, and Root Complex Event Collector (RCEC) logic. Local clusters map the detected errors to three error classes and report them to the global error logic. These errors are sorted into fatal, non-fatal, and correctable, and reported to the respective global error status registers. When an error is reported by the local cluster, the corresponding bit in the Global Fatal, Non-Fatal or Correctable Error Status register is set. The software clears the error bit by writing 1 to the bit. Each error is individually masked by the global error control registers. If an error is masked, the corresponding status bit is not set for any subsequent reported error. The Global Error Mask register is non-sticky and cleared by reset.

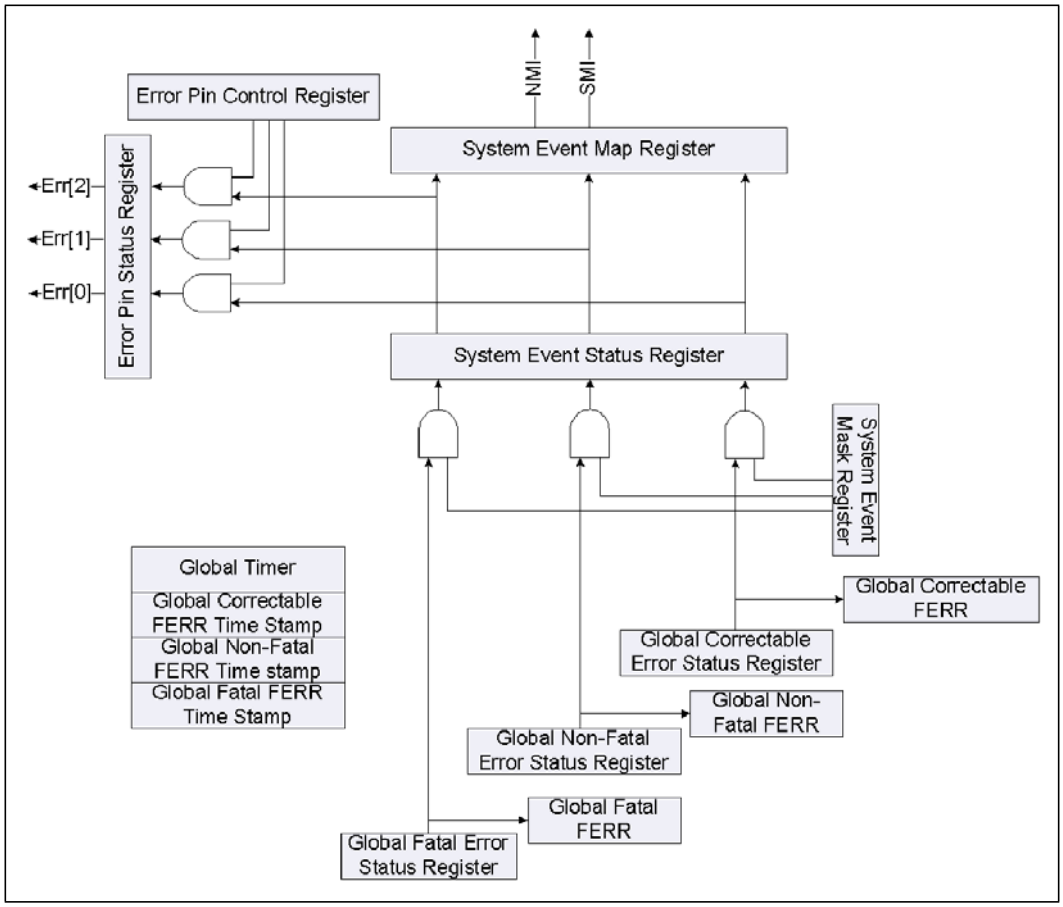
- Global Log Registers

  The Global Error Log registers log the errors reported by the SoC clusters. Local clusters map the detected errors to three error classes and report them to the global error logic. The three error classes are divided into fatal, non-fatal and correctable errors that are logged separately by the FERR and NERR registers. Each bit in the FERR/ NERR register is associated with a specific interface/cluster (e.g., a PCIe Root Port). Each bit is individually cleared by writing 1 to the bit. FERR logs the first report of an error, while NERR logs the subsequent reports of the other errors. The time stamp provides the time of when the first error was logged. The software reads this register to find out which of the local interfaces have reported the error. The FERR log remains valid and unchanged from the first error detection until the clearing of the corresponding error bit in the FERR by the software.

- Global System Event Register

  The errors collected by the global error registers are mapped to system events. The System Event Status bit reflects the logical OR output of all associated error severity unmasked errors. Each System Event Status bit individually masks by the System Event Control registers. Masking a System Event Status bit forces the corresponding bit to 0. When a System Event Status bit transitions from 0 to 1, the bit triggers one or more system events based on the programming of the System Event Map register as shown in Figure 4-7. Each error class is associated with one of the system events: SMI or NMI.

Figure 4-7. System Event Generation



In addition, the Error Pin registers allow error-pin assertion for an error. When an error is reported to the SoC, the SoC uses the severity level associated with the error to lookup for which system event is sent to the system. For example, a fatal error is mapped to an NMI with the ERROR2_B pin enabled by the software. If a fatal error is reported and logged by the Global Log register, then an NMI is dispatched to the CPU and the SoC asserts the ERROR2_B signal pin. The CPU or BMC reads the Global and Local Error Log register to determine where the error came from and how it handles the error.

At power-on reset, these registers are initialized to their default values. The default mapping of error class (severity) and system event is set to be consistent with Table 4-11. The firmware chooses to use the default values or modify the mapping according to the system requirements.

The System Event Mask register is a non-sticky register that is cleared by a hard reset.

Table 4-11. Default Error Severity Map

Error Severity	Error Reporting to CPU (Programmable)	Error Reporting to an External Device
Correctable Error	CPU: NMI/SMI - Default: SMI	ERROR0_B
Non-Fatal Error	CPU: NMI/SMI - Default: NMI	ERROR1_B
Fatal Error	CPU: NMI/SMI - Default: NMI	ERROR2_B

4.7.7.3 System Error (SERR)

A System Error (SERR) is generated by an SoC logic block to indicate a condition of serious system instability. The SERR events are mapped to an NMI or SMI at the SoC level.

4.7.7.4 First and Next Error Log Registers

This section describes local and global error logging. The log registers are named xxxxFERR and xxxxNERR where xxxx varies. First and next errors are captured at both the local level (correctable and uncorrectable) and the global level (correctable, non-fatal, and fatal). PCIe specifies its own error-logging mechanism which is not described here. Refer to the *PCI Express Base Specification*, Revision 2.1 for details.

For global error logging, the SoC categorizes the detected errors into fatal, non-fatal, and correctable based on the error severity. Each category includes two sets of error logging: FERR and NERR.The FERR register stores the information associated with the first detected error, while NERR stores the information associated with the subsequent detected errors after the first error. Both FERR and NERR log the error status of the same format. They indicate errors that are detected by the SoC in the format bit vector with one bit assigned to each error. A first error event is indicated by setting the corresponding bit in the FERR status register, a subsequent error is indicated by setting the corresponding bit in the NERR register. In addition, the local FERR registers also log the header of the erroneous cycle. Both first error and next error trigger system events.

Once the first error and the next error have been indicated and logged, the log registers for that error remains valid until either:

- The First Error bit is clear in the associated error status register, or
- The SoC generates the power-good, platform reset (PMU_PLTRST_B output pin).

The software clears an error bit by writing 1 to the corresponding bit position in the error status register.

The SoC hardware rules for updating the FERR and NERR registers and error logs are:

1. The first error event is indicated by setting the corresponding bit in the FERR status register, a subsequent error is indicated by setting the corresponding bit in the NERR status register.
2. If the same error occurs before the FERR status register bit is cleared, the error is not logged in the NERR status register.
3. In the case of simultaneous multiple errors with same severity, any two errors are logged in the FERR and NERR registers.
4. Updates to the error status and error log registers appear atomic to the software.
5. Once the first error information is logged in the FERR log register, the logging of the FERR log registers is disabled until the corresponding FERR error status is cleared by the software.
6. The error status registers, the error mask registers, and the error log registers are cleared by the power-on reset only. The contents of error log registers are preserved across a reset (while the power-good COREPWROK input pin remains asserted).

4.7.7.5 Error Register Flow

1. Upon a detection of a local error, the corresponding local error status is set if the error is unmasked; otherwise, the error bit is not set and the error is not propagated.
2. The local uncorrectable error is mapped to its associated error severity defined by the Uncorrectable Error Severity register. Setting the local error status bit causes the logging of the error—fatal/non-fatal errors are logged in the local uncorrectable FERR/NERR registers, while correctable errors are logged in the local correctable FERR/NERR registers. PCIe errors are logged according to the PCI Express Base Specification, Revision 2.1.
3. The local FERR and NERR logging events are forwarded to the global FERR and NERR registers. The report of local FERR/NERR sets the corresponding global error bit if the global error is unmasked; otherwise, the global error bit is not set and the error is propagated. The global FERR logs the first occurrence of local FERR/NERR event in the SoC, while the global NERR logs the subsequent local FERR/NERR event.
4. A correctable error is logged in the global correctable FERR/NERR registers, a non-fatal error is logged in the global non-fatal FERR/NERR registers, and a fatal error is logged in the global fatal FERR/NERR registers.
5. The Global Error register reports the error with its associated error severity to the System Event Status register. The system event status is set if the system event reporting is unmasked for the error severity; otherwise, the bit is not set and the error is not reported.
6. Setting the system event bit triggers a system event generation according the mapping defined in the System Event Map register. The associated system event is generated for the error severity and dispatched to the CPU/BMC of the error (interrupt for the CPU or error pin for the BMC).
7. The Global Log and Local Log registers provide the information to identify the source of the error. The software reads the log registers and clears the global and local error status bits.
8. Since the error status bits are edge triggered, a 0-to-1 transition is required to set the bit again. While the error status bit (local, global, or system event) is set to 1, all incoming error reporting to the respective error status register is ignored (no 0-to-1 transition).
    a. When a write to clear the local error status bit is done, the local error register re-evaluates the logical OR output of its error bits and reports it to the Global Error register; however, if the global error bit is already set, then the report is ignored.
    b. When a write to clear the error status bit is done, the Global Error register re-evaluates the logical OR output of its error bits and reports it to the System Event Status register; however, if the system event status bit is already set, then the report is not generated.
    c. The software optionally masks or unmasks the system event generation (interrupt or error pin) for an error severity in the System Event Control register while clearing the Local and Global Error registers.

9. The software has the following options for clearing the error status registers:
   a. Read the Global and Local Log registers to identify the source of the error. Clear the local error bits; this does not cause a generation of an interrupt with the global bit still set. Then, clear the global error bit and write to the local error register again with all 0s. Writing 0s to the local status does not clear any status bit, but causes the re-evaluation of the error status bits. An error is reported if any local error bit is unclear.
   b. Read the Global and Local Log registers to identify the source of the error and mask the error reporting for the error severity. Clear the system event and global error status bits—this causes setting of the system event status bit if the other global bits are still set. Then, clear the local error status bits—this causes setting of the global error status bit if other local error bits are still set. Then, unmask the system event to cause the SoC to report the error.
10. FERR logs the information of the first error detected by the associated error status register (local or global). The FERR log remains unchanged until all bits in the respective error status register are cleared by the software. When all error bits are cleared, the FERR logging is re-enabled.

4.7.7.6 Error Counters

This feature allows the system management controller to monitor the component health by periodically reporting the correctable error count. The error RAS structure already provides a first-error status and a second-error status. Because the response time of system management is on the order of milliseconds, reading and clearing the error logs in time to detect short bursts of errors across the SoC component does not happen. Over a long period of time, the software uses these values to monitor the rate of change in the error occurrences. This helps identify potential component degradations, especially with respect to the memory interface.

A register with one-hot encoding selects which error types participate in error counting. More than one error is unlikely to occur within a cluster at a given time. The SoC only counts one occurrence in one clock cycle. The selection register logically ORs-together all of the selected error types to form a single count enable. This means that only one counter increment occurs for one or all types selected. Register attributes are set to write a 1 to clear.

Each cluster has one set of error counter/control registers.

- The SMBus device contains one 7-bit counter (SMBus_ERRCNT[6:0]).
  - Bit[7] is an overflow bit, all bits are sticky with a write logic 1 to clear.
- The root complex device contains one 7-bit counter (RTF_ERRCNT[6:0]).
  - Bit[7] is an overflow bit, all bits are sticky with a write logic 1 to clear.
- The internal memory controller contains one 7-bit counter (Dunit_ERRCNT[6:0]).
  - Bit[7] is an overflow bit, all bits are sticky with a write logic 1 to clear.
- The SoC system agent contains one 7-bit counter (Bunit_ERRCNT[6:0]).
  - Bit[7] is an overflow bit, all bits are sticky with a write logic 1 to clear.

4.8 SoC Error Handling Summary

The following tables provide a summary of the errors that are monitored by the SoC. The errors are reported to the CPU and/or to an external device (e.g., BMC).

Table 4-12 shows the default error severity mapping in the SoC and how each error severity is reported, while Table 4-13 summarizes the default logging and responses on the SoC detected errors.

Table 4-12. Default Error Severity

Error Severity	Error Reporting to CPU (Programmable)	Error Reporting to a Device External to the SoC
Correctable Error	CPU: NMI/SMI Default: SMI	ERROR0_B
Non-Fatal Error	CPU: NMI/SMI Default: NMI	ERROR1_B
Fatal Error	CPU: NMI/SMI Default: NMI	ERROR2_B

Table 4-13. Summary of Default Error Logging and Responses (Sheet 1 of 6)

ID	Error	Error Type (Default Severity)	Transaction Response	Default Error Logging[1]
I/O Fabric Errors				
A0	RTF Detected Command Parity Error	Uncorrectable (Fatal)	Internal I/O Bus Command Parity errors are detected at the output of the upstream and downstream queues when a request is granted. Further arbiter grants are inhibited.	FERR/NERR is logged in RTF and Global Fatal Error Log registers: RTF_ERRUNCSTS RTF_FERRUNCSTS RTF_NERRUNCSTS RTF_FERRHDRLOG GFERRSTS GFFERRSTS GFNERRSTS GFFERRTIME Header is logged.
A1	Cfg Agent Detected IOSF Data Parity Error		SoC detects and logs the error.	
A2	Cfg Agent Detected Configuration Register Parity Error		SoC detects and logs the error.	
A3	B-Unit to FNB Data Parity		SoC detects and logs the error.	
A4	FNB to B-Unit Data Parity		SoC logs the error, corrupts parity for all bytes chunks associated with this data logs all 1s as header.	
A5	FNB to B-Unit Write Header Parity		SoC logs the error and the header.	
A6	FNB to B-Unit Read Header Parity		SoC logs the error and the header.	
A7	FNB to A-Unit Data Parity		SoC logs the error.	
A8	FNB to A-Unit Header Parity		SoC logs the error and the header.	
B-Unit Errors				
B0	SSA BRAM Data Parity Error	Uncorrectable (Fatal)	SoC detects and logs the error.	FERR/NERR is logged in B-Unit and Global Fatal Error Log registers: GFERRSTS GFFERRSTS GFNERRSTS GFFERRTIME
B1	P-Unit to SSA Data Parity Error			
B2	FNB to SSA Data Parity Error			
B3	D-Unit to SSA Data Parity Error			

Table 4-13. Summary of Default Error Logging and Responses (Sheet 2 of 6)

ID	Error	Error Type (Default Severity)	Transaction Response	Default Error Logging[1]
GbE Errors				
C0	Receiver Error (RE)	Correctable		Error is logged in GbE Local and Global Correctable Error Log registers: CES GCORERRSTS GCORFERRSTS GCORNERRSTS GCORFERRTIME
C1	Bad TLP Error (BTLPE)			
C2	Bad DLLP Error (BDLLPE)			
C3	Replay Number Rollover Error (RNRE)			
C4	Replay Timer Timeout Error (RTTE)			
C5	Advisory Non-Fatal Error (ANFE)			
C6	Poisoned TLP Error (PTLPE)	Uncorrectable (Non-Fatal)	ERR_Non-fatal sent to the root complex. Header is logged. A poisoned completion is ignored and the request can be retried after timeout. If enabled, the error is reported.	Error is logged in the GbE Local and Global Non-fatal Error Log registers: UES GNERRSTS GNFERRSTS GNNERRSTS GNFERRTIME
C7	Completion Timeout Error (CTE)		Error severity is non-fatal (default case): Send error message. If advisory, retry the request once and send the advisory error message on each failure. If fails, send uncorrectable error message. Error severity is defined as fatal: Send uncorrectable error message.	
C8	Completer Abort Error (CAE)		ERR_Non-fatal sent to the root complex. Header is logged. Send completion with CA.	
C9	Unexpected Completion Error (UCE)		ERR_Non-fatal sent to the root complex. Header is logged. Discard TLP.	
C10	ECRC Error (EE)			
C11	Unsupported Request Error (URE)		ERR-Non-fatal sent to the root complex. Header is logged. Send completion with a UR.	
C12	Data Link Protocol Error (DLPE)	Uncorrectable (Fatal)		Error is logged in the GbE Local and Global Fatal Error Log registers: UES GFERRSTS GFFERRSTS GFNERRSTS GFFERRTIME
C13	Flow Control Error (FCE)			
C14	Receiver Overflow Error (ROE)			
C15	Malformed TLP Error (MTLPE)		ERR_Fatal sent to the root complex. Header logged. Drop the packet and free FC credits.	

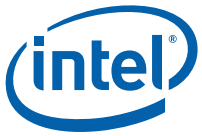


Table 4-13. Summary of Default Error Logging and Responses (Sheet 3 of 6)

ID	Error	Error Type (Default Severity)	Transaction Response	Default Error Logging[1]
Memory Controller Channel 1 Errors				
D1_0	Read Data Correctable ECC Error	Correctable	Read data is corrected and sent to the B-Unit.	FERR/NERR is logged in D-Unit and Global Fatal Error Log registers: DERRSTS FERRNERR SBELOG UCELOG GFERRSTS GFFERRSTS GFNERRSTS GFFERRTIME
D1_1	Write Data Parity Error	Uncorrectable (Fatal)	Write data is written with poisoned ECC.	
D1_2	Read Data Uncorrectable ECC Error (address error, all four ECC syndromes are equal and correspond to an address error)		Read data is sent with bad parity to the B-Unit.	
D1_3	Read Data Uncorrectable ECC Error (general - All other non-zero ECC syndromes)		Read data is sent with bad parity to the B-Unit.	
Memory Controller Channel 2 Errors				
D2_0	Read Data Correctable ECC Error	Correctable	Read data is corrected and sent to the B-Unit.	FERR/NERR is logged in D-Unit and Global Fatal Error Log registers: DERRSTS FERRNERR SBELOG UCELOG GFERRSTS GFFERRSTS GFNERRSTS GFFERRTIME
D2_1	Write Data Parity Error	Uncorrectable (Fatal)	Write data is written with poisoned ECC.	
D2_2	Read Data Uncorrectable ECC Error (address error, all four ECC syndromes are equal and correspond to an address error)		Read data is sent with bad parity to the B-Unit.	
D2_3	Read Data Uncorrectable ECC Error (general - All other non-zero ECC syndromes)		Read data is sent with bad parity to the B-Unit.	

Table 4-13. Summary of Default Error Logging and Responses (Sheet 4 of 6)

ID	Error	Error Type (Default Severity)	Transaction Response	Default Error Logging[1]
PCIe* Root Port Errors				
E0	Receiver Error (RE)	Correctable	Respond per *PCI-E Specification*	Log error per PCI Express* AER requirements for these correctable errors/message. If the PCIe correctable error is forwarded to the Global Error registers, the error is logged in Global Correctable Log registers: GCORERRSTS GCORFERRSTS GCORNERRSTS GCORFERRTIME
E1	Bad TLP Error (BTLPE)			
E2	Bad DLLP Error (BDLLPE)			
E3	Replay Number Rollover Error (RNRE)			
E4	Replay Timer Time-out Error (RTTE)			
E5	Header Log Overflow Error (HLOE)			
E6	Received ERR_COR Message from Downstream Device			
E7	Poisoned TLP Error (PTLPE)	Uncorrectable (Non-fatal)	SoC logs the error.	Log error per PCI Express AER requirements for the corresponding error/message. If the PCIe uncorrectable error is forwarded to the Global Error registers, the error is logged in Global Non-Fatal Log registers: GNERRSTS GNFERRSTS GNNERRSTS GNFERRTIME
E8	Completion Time-out Error (CTE)			
E9	Completer Abort Error (CAE)			
E10	Unexpected Completion Error (UCE)		Respond Per *PCI-E Specification*	
E11	Unsupported Request Error (URE)		SoC logs the error.	
E12	ACS Violation Error (ACSE)			
E13	MC Blocked TLP Error (MCE)			
E14	Atomic Egress Blocked Error (AEBE)			
E15	Received ERR_NONFATAL Message from Downstream Device		Respond per *PCI-E Specification*	
E16	Data Link Protocol Error (DLPE)	Uncorrectable (Fatal)	Respond per *PCI-E Specification*	Log error per PCI Express AER requirements for the corresponding error/message. If the PCIe uncorrectable error is forwarded to the Global Error registers, the error is logged in Global Fatal Log registers: GFERRSTS GFFERRSTS GFNERRSTS GFFERRTIME
E17	Surprise Link Down Error (SLDE)			
E18	Flow Control Error (FCE)			
E19	Receiver Overflow Error (ROE)			
E20	Malformed TLP Error (MTLPE)			
E21	Uncorrectable Internal Error (UIE)			
E22	Received ERR_FATAL Message from Downstream Device			

Table 4-13. Summary of Default Error Logging and Responses (Sheet 5 of 6)

ID	Error	Error Type (Default Severity)	Transaction Response	Default Error Logging[1]
SMBus Errors				
F0	Retry Error (RETRYERR): An error due to SMT master transaction exceeding (non-collision) retry count as specified in RPOLICY.RETRY	Uncorrectable (Fatal)	SoC detects and logs the error.	FERR/NERR is logged in the Global Fatal Error Log registers: GFERRSTS GFFERRSTS GFNERRSTS GFFERRTIME
SATA2 Errors				
H0	When a set event occurs to configuration bits SATAGC.URD=1, during SATAGC.URRE=1 and CMD.SEE=1 for controller 1.	Uncorrectable (Fatal)	SoC detects and logs the error.	Error is logged in PCISTS and Global Fatal Error Log registers: PCISTS GFERRSTS GFFERRSTS GFNERRSTS GFFERRTIME
H1	When a set event occurs to configuration bits STS.DPE=1 during CMD.PEE=1 and CMD.SEE=1 for controller 1.			
H2	When a set event occurs to configuration bits STS.STA=1 during CMD.SEE=1 for controller 1.			
SATA3 Errors				
I0	When a set event occurs to configuration bits SATAGC.URD=1, during SATAGC.URRE=1 and CMD.SEE=1 for controller 1.	Uncorrectable (Fatal)	SoC detects and logs the error.	Error is logged in PCISTS and Global Fatal Error Log registers: PCISTS GFERRSTS GFFERRSTS GFNERRSTS GFFERRTIME
I1	When a set event occurs to configuration bits STS.DPE=1 during CMD.PEE=1 and CMD.SEE=1 for controller 1.			
I2	When a set event occurs to configuration bits STS.STA=1 during CMD.SEE=1 for controller 1			

Table 4-13. Summary of Default Error Logging and Responses (Sheet 6 of 6)

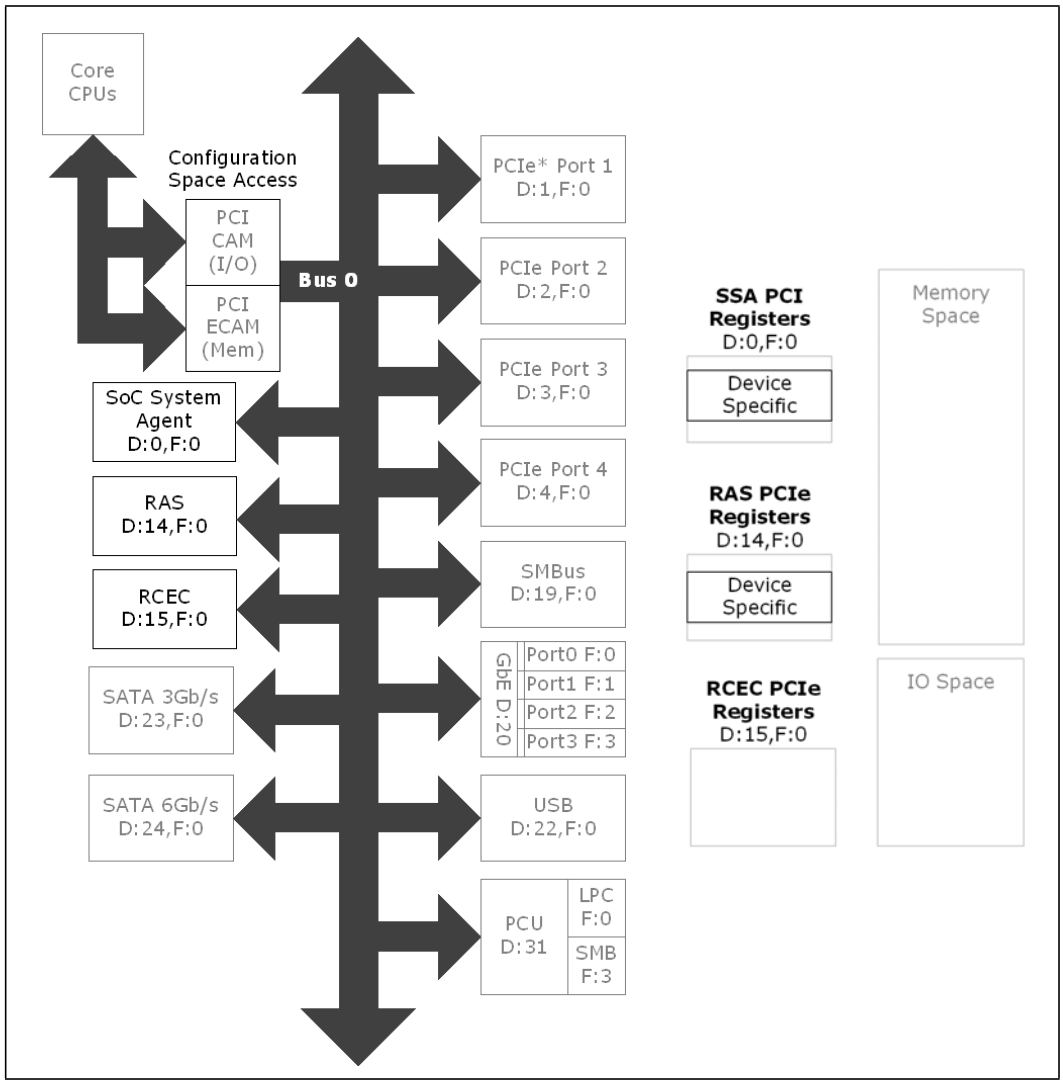
ID	Error	Error Type (Default Severity)	Transaction Response	Default Error Logging[1]
USB Errors				
J0	Data Parity Error	Uncorrectable (Fatal)	SoC detects and logs the error.	Error is logged in PCISTS and Global Fatal Error Log registers: PCISTS GFERRSTS GFFERRSTS GFNERRSTS GFFERRTIME
J1	Address or Command Parity Error			
J2	Reception of status other than "Successful" on a memory read completion			
Platform Controller Unit (PCU) Errors				
K0	LPC Sync Error	Uncorrectable (Fatal)	SoC detects and logs the error.	Error is logged in PCISTS and Global Fatal Error Log registers: PCISTS GFERRSTS GFFERRSTS GFNERRSTS GFFERRTIME

1. This column notes the logging registers used assuming the error severity default remains. The error severity dictates the actual logging registers used upon detecting an error.

4.9 Register Map

Figure 4-8 shows the SSA, RAS, and RCEC register map.

Figure 4-8. Register Map



4.10 System Agent Register Map

4.10.1 Registers in Configuration Space

Bus 0; Device 0; Function 0

Table 4-14. Header Registers

Offset	Name	Description
000h	CUNIT_REG_DEVICEID	CUNIT_REG_DEVICEID
004h	CUNIT_CFG_REG_PCISTATUS	CUNIT_CFG_REG_PCISTATUS
008h	UNIT_CFG_REG_CLASSCODE	UNIT_CFG_REG_CLASSCODE
00Ch	CUNIT_CFG_REG_HDR_TYPE	CUNIT_CFG_REG_HDR_TYPE

Table 4-15. Device Specific Registers

Offset	Name	Description
0D0h	CUNIT_MSG_CTRL_REG	CUNIT_MSG_CTRL_REG
0D4h	CUNIT_MSG_DATA_REG	CUNIT_MSG_DATA_REG
0D8h	CUNIT_MSG_CTRL_REG_EXT	CUNIT_MSG_CTRL_REG_EXT
0DCh	CUNIT_MSG_CTRL_PACKET_REG	CUNIT_MSG_CTRL_PACKET_REG
0F0h	CUNIT_SCRATCHPAD_REG	CUNIT_SCRATCHPAD_REG
0F8h	CUNIT_MANUFACTURING_ID	CUNIT_MANUFACTURING_ID
100h	CUNIT_LOCAL_CONTROL_MODE	CUNIT_LOCAL_CONTROL_MODE
104h	CUNIT_ACCESS_CTRL_VIOL	CUNIT_ACCESS_CTRL_VIOL
108h	CUNIT_PDM_REGISTER	CUNIT_PDM_REGISTER
114h	CUNIT_MCRS_SAI	CUNIT_MCRS_SAI
118h	CUNIT_MDR_SAI	CUNIT_MDR_SAI

4.11 RAS Register Map

4.11.1 Registers in Configuration Space

Bus 0; Device 14 (decimal); Function 0

Table 4-16. Device Specific - Global Error Registers

Offset	Name	Description
200h	GCORERRSTS	Global Correctable Error Status Register
204h	GNERRSTS	Global Non-Fatal Error Status Register
208h	GFERRSTS	Global Fatal Error Status Register
20Ch	GERRMSK	Global Error Mask Register
210h	GCORFERRSTS	Global Correctable FERR Status Register
214h	GCORNERRSTS	Global Correctable NERR Status Register
218h	GNFERRSTS	Global Non-Fatal FERR Status Register
21Ch	GNNERRSTS	Global Non-Fatal NERR Status Register
220h	GFFERRSTS	Global Fatal FERR Status Register
224h	GFNERRSTS	Global Fatal NERR Status Register
228h	GTIME	Global Error Timer Register
230h	GCORFERRTIME	Global Correctable FERR Error Time Stamp Register
238h	GNFERRTIME	Global Non-Fatal FERR Error Time Stamp Register
240h	GFFERRTIME	Global Fatal FERR Error Time Stamp Register
248h	GSYSEVTSTS	Global System Event Status Register
24Ch	GSYSEVTMSK	Global System Event Mask Register
250h	GSYSEVTMAP	Global System Event Map Register
254h	ERRPINCTRL	Error Pin Control Register
258h	ERRPINSTS	Error Pin Status Register
25Ch	ERRPINDATA	Error Pin Data Register

Table 4-17. Device Specific - Root Complex Local Error Registers

Offset	Name	Description
280h	RTF_ERRUNCSTS	RTF Uncorrectable Error Status Register
284h	RTF_ERRUNCMSK	RTF Uncorrectable Error Mask Register
288h	RTF_FERRUNCSTS	RTF Uncorrectable First Error Status Register
28Ch	RTF_NERRUNCSTS	RTF Uncorrectable Next Error Status Register
290h	RTF_UNCERRCNTSEL	RTF Uncorrectable Error Counter Selection Register
294h	RTF_UNCERRCNT	RTF Uncorrectable Error Counter Register
298h	RTF_FERRHDRLOG1	Header Log Register 1
29Ch	RTF_FERRHDRLOG2	Header Log Register 2
2A0h	RTF_FERRHDRLOG3	Header Log Register 3
2A4h	RTF_FERRHDRLOG4	Header Log Register 4

Table 4-18. Device Specific - Fabric Configuration Registers

Offset	Name	Description
404h	RTF_BMBOUND	RTF BMBOUND Register
408h	RTF_BMBOUNDHI	RTF BMBOUNDHI Register
40Ch	RP_BIFCTL	Root Port Bifurcation Control Register

4.12 Root Complex Event Collector (RCEC) Register Map

4.12.1 Registers in Configuration Space

Bus 0; Device 15 (decimal); Function 0

Base Class (BC) = 08h: Root Complex Event Collector

Table 4-19. PCI Standard Type 0 Header Registers

Offset	Name	Description
00h	VID	Vendor ID Register
02h	DID	Device ID Register
04h	PCICMD	PCI Command Register
06h	PCISTS	PCI Status Register
08h	RID	Revision ID Register
09h	CC	Class Code Register
0Ch	CLS	Cacheline Size Register
0Eh	HDR	Header Type Register
2Ch	SVID	Subsystem Vendor ID Register
2Eh	SID	Subsystem ID Register
34h	CAPPTR	Capabilities Pointer Register
3Ch	INTL	Interrupt Line Register
3Dh	INTP	Interrupt Pin Register

Table 4-20. PCI Express Capability Structure

Offset	Name	Description
40h	EXPCAPLST	PCI Express* Capability List Register
42h	EXPCAP	PCI Express Capabilities Register
44h	DEVCAP	Device Capabilities Register
48h	DEVCTL	Device Control Register
4Ah	DEVSTS	Device Status Register
5Ch	ROOTCTL	Root Control Register
5Eh	ROOTCAP	Root Capabilities Register
60h	ROOTSTS	Root Status Register

Table 4-21. Power Management Capability Structure

Offset	Name	Description
80h	PMCAPLST	Power Management Capability List Register
82h	PMCAP	Power Management Capabilities Register
84h	PMCSR	Power Management Control / Status Register

Table 4-22. MSI Capability Structure

Offset	Name	Description
90h	MSICAPLST	MSI Capability List Register
92h	MSICTL	MSI Message Control Register
94h	MSIADDR	MSI Message Address Register
98h	MSIDATA	MSI Message Data Register
9Ch	MSIMSK	MSI Mask Bit Register
A0h	MSIPENDING	MSI Pending Bit Register

Table 4-23. Advanced Error Reporting (AER)

Offset	Name	Description
100h	AERCAPHDR	Advanced Error Reporting Extended Capability Header
104h	ERRUNCSTS	Uncorrectable Error Status Register
108h	ERRUNCMSK	Uncorrectable Error Mask Register
10Ch	ERRUNCSEV	Uncorrectable Error Severity Register
110h	ERRCORSTS	Correctable Error Status Register
114h	ERRCORMSK	Correctable Error Mask Register
118h	AERCAPCTL	Advanced Error Capabilities and Control Register
11Ch	AERHDRLOG1	Header Log Register 1
120h	AERHDRLOG2	Header Log Register 2
124h	AERHDRLOG3	Header Log Register 3
128h	AERHDRLOG4	Header Log Register 4
12Ch	ROOTERRCMD	Root Error Command Register
130C	ROOTERRSTS	Root Error Status Register
134h	ERRSRCID	Error Source Identification Register

Table 4-24. Root Complex Event Collector Endpoint Association

Offset	Name	Description
150h	RCECEPACAPHDR	RCEC Endpoint Association Extended Capability Header
154h	ABMRCIEP	Association Bitmap for Root Complex Integrated Endpoints Register

§ §

5 Clock Architecture

The SoC contains a variable frequency, multiple clock domain, and a multiple power plain clocking system. The architecture includes a clock synchronization scheme, a multiple clock domain crossing, management of skew between the CPU and the rest of the system, and consolidation of PLLs.

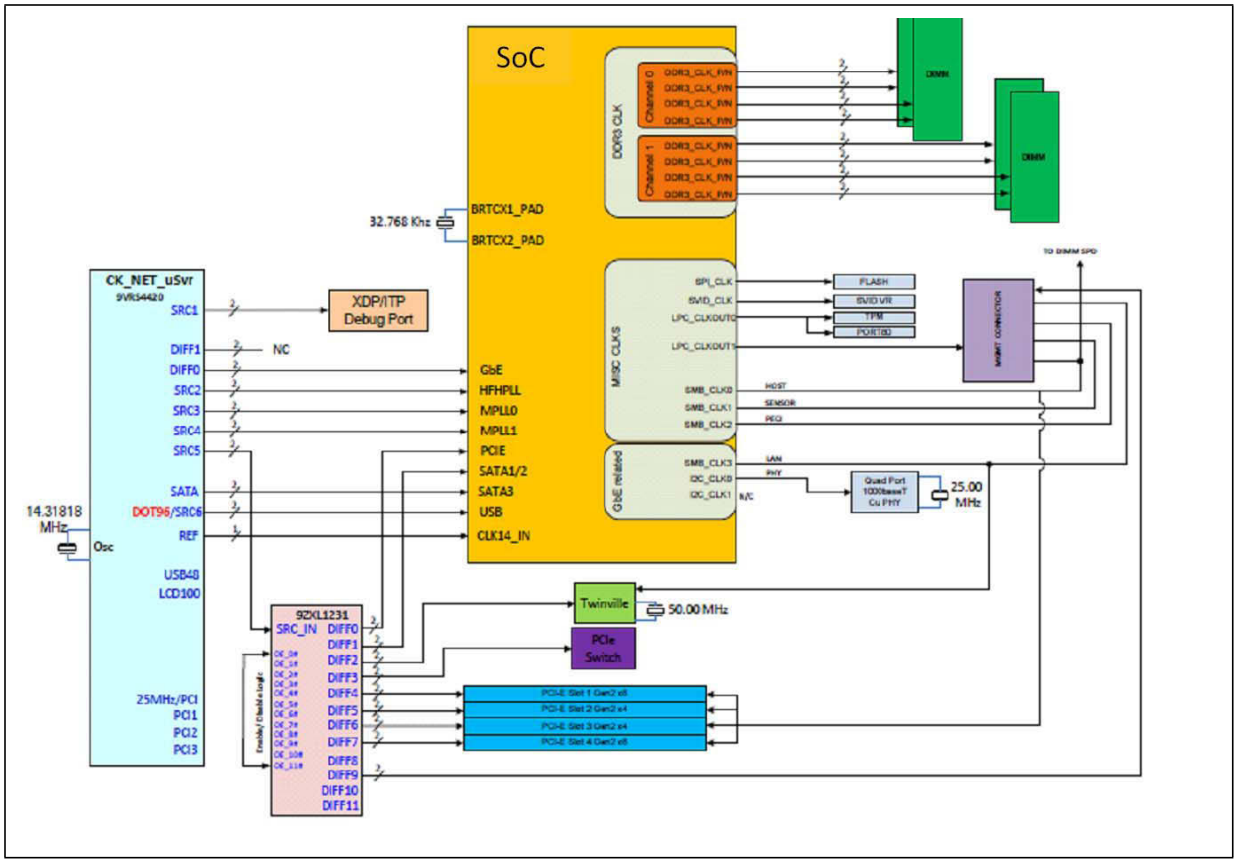
A clock synthesizer component can be used to provide the reference clocks to the SoC. The Customer Reference Board (CRB) designs associated with the SoC product use a clock synthesizer component marketed by Integrated Device Technology, Inc*.

The reference clocks must comply with the specifications given in Section 33.16, "SoC Reference Clock Interfaces" on page 668. Some require Spread-Spectrum Clocking (SSC). The SoC reference-clock inputs are:

- 100 MHz differential, *PCI Express* 2.0 Specification* compliant, SSC required
- 100 MHz differential isolated for SATA 2, SSC required
- 100 MHz differential isolated for SATA 3, either SSC or non-SSC can be used
- 100 MHz differential, for memory controller 0, SSC required
- 100 MHz differential, for memory controller 1, SSC required
- 100 MHz differential, for host PLL, SSC required
- 100 MHz (or 125 MHz for 2.5 GbE) differential, for GbE controller, non-SSC
- 96 MHz differential, for USB controller, non-SSC
- 14.318 MHz single-ended, non-SSC

Figure 5-1 shows the clock architecture as implemented on a CRB.

Figure 5-1. Clock Architecture



5.1 Input Clocks

Table 5-1. Input Clocks

Signal	Frequency	Usage
GBE_REFCLKN, GBE_REFCLKP	100 MHz or 125 MHz	GbE 100/125 MHz differential clock. External SerDes/ SGMII differential 100/125 MHz reference clock from an external generator. 125 MHz is required for 2.5 GbE operation.
HPLL_REFN, HPLL_REFP	100 MHz	100 MHz differential reference clock from an external generator
DDR3_0_REFP, DDR3_0_REFN	100 MHz	DDR3 reference clock at 100 MHz for memory controller 0
DDR3_1_REFP, DDR3_1_REFN	100 MHz	DDR3 reference clock at 100 MHz for memory controller 1
PCIE_REFCLKN, PCIE_REFCLKP	100 MHz	PCIE reference clock
SATA_REFCLKP, SATA_REFCLKN	100 MHz	SATA reference clock
SATA3_REFCLKP, SATA3_REFCLKN	100 MHz	SATA3 reference clock
USB_REFCLKP, USB_REFCLKN	96 MHz	USB reference clock
CLK14_IN	14.31838 MHz	Clock-to-legacy block

5.2 Output Clocks

Table 5-2. Output Clocks

Signal	Frequency	Description
DDR3_0_CK[3:0]	1333, 1600 MHz	Output to DIMMs
DDR3_0_CKB[3:0]	1333, 1600 MHz	Output to DIMMs
DDR3_1_CK[3:0]	1333, 1600 MHz	Output to DIMMs
DDR3_1_CKB[3:0]	1333, 1600 MHz	Output to DIMMs
SPI_CLK	20/33 MHz	Clock supplied to external Flash device and toggles only when a transaction is going over the SPI interface.
SVID_CLK	11 – 25 MHz	Clock used by voltage regulator
SMB_CLK0 / GPIOS_9	100 kHz	SMBus clock wire
SMB_CLK1/GPIO_12	100 kHz	SMBus clock wire
SMB_CLK2 / GPIOS_14	100 kHz	SMBus clock wire
GBE_SMBCLK/ NCSI_CLK_IN	84 kHz – SMBus Master 50 MHz when SoC provides NCSI_CLK_OUT	Clock used by BMC interface
GBE_MDIO0_I2C_CLK	Clock for GBE I^2C interface	
GBE_MDIO1_I2C_CLK	Clock for GBE I^2C interface	
LPC_CLKOUT0 LPC_CLKOUT1	25 MHz	Provided to devices requiring LPC clock
FLEX_CLK_SE0 FLEX_CLK_SE1	25/33 MHz	Single-ended flexible clock Single-ended flexible clock
PMU_SUSCLK	32.768 kHz	Suspend clock

§ §

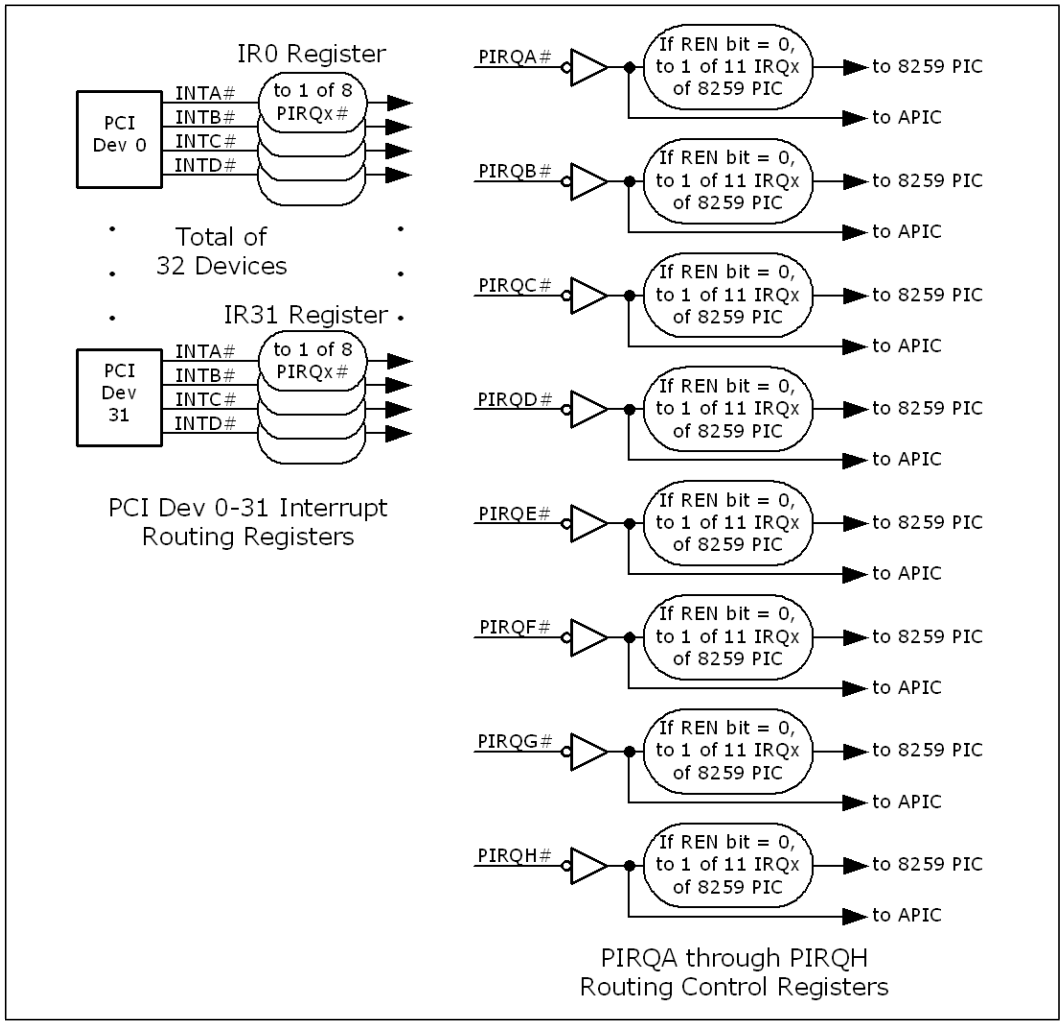
6 Interrupt Architecture

Global interrupt handling is described in this chapter. The types of interrupts are described as well as the routing and mapping of these interrupts to the integrated I/O Advanced Programmable Interrupt Controller (APIC) and 8259 Programmable Interrupt Controller (PIC).

6.1 PCI Interrupts and Routing

The SoC does not provide external INTA# through INTD# PCI interrupt signal pins. The PCI Express* integrated endpoints and the PCIe* interrupt messages internally produce the equivalent of these four interrupt signals for each device. For each of the possible 32 PCI devices, each of these four interrupt signals can be programmed to be routed to one of eight internal interrupt signals called PIRQA# through PIRQH#. For a diagram of the routing scheme, see Figure 6-1.

Figure 6-1. PCI Interrupt Routing



There are 32, 16-bit PCI interrupt routing registers, one for each of the 32 possible PCI device addresses. These 32 registers, named IR0 through IR31, are located in memory space and begin at the memory address specified by ILB_BASE_ADDRESS, offset 20h.

Each 16-bit register has four 4-bit fields:

- IRA: INTA# mapping to one of PIRQA# through PIRQH# (0h through 7h)
- IRB: INTB# mapping to one of PIRQA# through PIRQH# (0h through 7h)
- IEC: INTC# mapping to one of PIRQA# through PIRQH# (0h through 7h)
- IRD: INTD# mapping to one of PIRQA# through PIRQH# (0h through 7h)

The field values 8h through Fh are reserved and must not be used.

Note: Each of the eight PIRQx# can represent the interrupt of more than one device and up to four for each device.

PIRQA# through PIRQH# are then each inverted and routed to the I/O APIC and to the 8259 PIC.

PIRQA through PIRQH are connected to the I/O APIC inputs IRQ16 through IRQ23, respectively. Information about programming the I/O APIC is in Chapter 30, "I/O Advanced APIC (I/O APIC)."

For connecting to the 8259 PIC, there are eight, 8-bit routing control registers, one for each PIRQA through PIRQH. Each of the eight interrupts are routed to one of 11 IRQ inputs of the integrated 8259 PIC according to its programmed IRQ Routing (IR) field. The PIRQx is routed to the 8259 PIC provided that its Interrupt Routing Enable (REN) bit is programmed to 0. When REN is programmed as 1, the particular PIRQx is not routed to the 8259 PIC. These eight registers, named PIRQA through PIRQH, are located in the memory space and begin at the memory address specified by the ILB_BASE_ADDRESS, offset 8h.

PIRQA# through PIRQH# are defined as Level-Sensitive interrupts. When routed to a specified IRQ line, the software must change the corresponding ELCR1 or ELCR2 register of the 8259 to a Level-Sensitive mode. Information about programming the 8259 PIC is in Chapter 29, "8259 Programmable Interrupt Controller (PIC)."

The PIRQx register decode is shown in Table 6-1.

Table 6-1. PIRQA through PIRQH Routing Register IRQ Decode

PIRQx Register, IR field 3:0	PIRQx Routing to 8259 PIC[1]
0000	Reserved (default)
0001	Reserved
0010	Reserved
0011	IRQ3 of the 8259 PIC
0100	IRQ4 of the 8259 PIC
0101	IRQ5 of the 8259 PIC
0110	IRQ6 of the 8259 PIC
0111	IRQ7 of the 8259 PIC
1000	Reserved
1001	IRQ9 of the 8259 PIC
1010	IRQ10 of the 8259 PIC
1011	IRQ11 of the 8259 PIC
1100	IRQ12 of the 8259 PIC
1101	Reserved
1110	IRQ14 of the 8259 PIC
1111	IRQ15 of the 8259 PIC

1. The REN bit (bit 7) must be set to 0 or else the PIRQx is not routed to the 8259 PIC. The default setting of REN =1.

6.2 Non-Maskable Interrupt (NMI)

NMI is generated by serious system events, memory parity errors, some System Errors (SERR), and PCI Express fatal errors.

The SoC provides an external input signal pin (NMI) and is detected on the rising edge. The NMI is reset by the software by setting the corresponding NMI source enable/disable bit in the NMI Status and Control (NSC) register. The NMI input pin is pin-muxed with the General Purpose I/O signal, GPIOS_0, and so must be pin-configured to be usable.

The 8-bit NSC register is located in the I/O space at 61h.

The NMI is broadcast to all cores in the SoC. Sleeping cores are first awakened.

The SoC provides interrupt mapping to generate the NMI for various SoC and system events. More information is provided in Chapter 21, "Intel Legacy Block (iLB) Devices."

6.3 System Management Interrupt (SMI)

The SMI indicates any of the several system-level conditions. Examples are:

- Thermal-sensor events
- Throttling activation
- System management RAM access
- Chassis open
- System power button pressed
- System Management Bus (SMBus) events
- Power Management Events (PME)
- PCI Express Hot-Plug* events (Hot-Plug events are not supported by the SoC.)
- Real Time Clock (RTC) alarm activation
- Various system-state-related activities

An SMI causes the system to enter System Management Mode (SMM). SMM is an operating mode in which all normal execution, including the operating system, is suspended, and special, separate software, usually firmware or a hardware-assisted debugger, is executed in high-privilege mode. The SMI is broadcast to all cores in the SoC. Sleeping cores are first awakened.

Various events can be programmed to generate either a System Control Interrupt (SCI) or as an SMI. Mapping events to SMI can be used when a legacy (APM) OS is in use. The SCI Enable (SCI_EN) bit of the Power Management 1 Control (PM1_CNT) register controls whether the event is routed as an SCI or an SMI. The PM1_CNT register is located in the I/O space at ACPI_BASE_ADDRESS, offset 4.

6.4 System Control Interrupt (SCI)

SCI is a special type of hardware power-management interrupt that is handled directly by the OS, and is not handled by a device driver. It is closely tied to the ACPI model. The operating system uses the SCI interrupt to process ACPI events signaled by GPEs, whether the system is asleep or awake when the event occurs.

SCI can be useful in cases where a driver is not loaded. For example, when a device has been placed in the D3 power state while the system remains in S0, or for the delivery of events such as power button to the OS.

If not using the I/O APIC for the SCI, the SCI must be routed to IRQ9-IRQ11 of the 8259 PIC. When routed to the 8259 PIC, the SCI is not sharable with the Serial Interrupt (SERIRQ) for that PIC input, but it is shareable with the PIRQA through PIRQH interrupts.

If using an I/O APIC, the SCI is mapped to the I/O APIC interrupt or to an SMI. Mapping SCI events to SMI can be used when a legacy (APM) OS is in use. The SCI Enable (SCI_EN) bit of the Power Management 1 Control (PM1_CNT) register controls whether the event is routed as an SCI or an SMI. The PM1_CNT register is located in the I/O space at ACPI_BASE_ADDRESS, offset 4.

The SCI routing to the I/O APIC is controlled by the 3-bit SCI IRQ Select (SCIS) field of the ACPI Control (ACTL) register located in the memory space at ILB_BASE_ADDRESSS, offset 0. See Table 6-2.

Also, if using the I/O APIC for SCI, the SCI can be mapped to IRQ20-IRQ23 of the I/O APIC, and can be shared with other interrupts. Here the SCI must be programmed for active-low reception. The SCI can also be mapped to IRQ9, IRQ10, of IRQ11 where it must be programmed for active-high reception.

Table 6-2. Routing of SCI to the I/O APIC

ACTL.SCIS	I/O APIC Input to Which the SCI is Routed
000	IRQ9
001	IRQ10
010	IRQ11
011	SCI Disabled
100	IRQ20
101	IRQ21
110	IRQ22
111	IRQ23

6.5 Message Signaled Interrupt (MSI and MSI-X)

MSI and MSI-X are generated by PCI and PCI Express devices that have this capability. During device configuration, each capable PCI function is allocated one or more vectors and the memory-mapped location to write the interrupt messages. When written in to memory, the MSI is communicated to the appropriate CPU through its local APIC.

6.6 I/O APIC Input Mapping

The SoC has an integrated I/O APIC which supports 24 APIC interrupts. Each interrupt has its own unique vector assigned by the software. The interrupt vectors are mapped as shown in Table 6-3.

Table 6-3. I/O APIC Input Mapping

I/O APIC Input	Interrupts Routed to This I/O APIC Input	Note
IRQ0	INTR output of the integrated 8259 PIC	
IRQ1	• SERIRQ (1), or • Keyboard (KBD) Emulation Interrupt	
IRQ2	• HPET Timer 0 (if GCFG.LRE is set) • 8254 Timer (if GCFG.LRE is not set)	1, 2, 3
IRQ3	• SERIRQ (3), or • UART COM2	
IRQ4	• SERIRQ (4), or • UART COM1	
IRQ5	• SERIRQ (5)	
IRQ6	• SERIRQ (6)	
IRQ7	• SERIRQ (7)	
IRQ8	• HPET Timer 1 (if GCFG.LRE is set) • RTC (if GCFG.LRE is not set)	1, 2, 3
IRQ9	• SERIRQ (9), or • from SCI (based on ACTL.SCIS and PM1_CNT.SCI_EN registers)	
IRQ10	• SERIRQ (10), or • from SCI (based on ACTL.SCIS and PM1_CNT.SCI_EN registers)	
IRQ11	• SERIRQ (11), or • HPET Timer 2, or • from SCI (based on ACTL.SCIS and PM1_CNT.SCI_EN registers)	4
IRQ12	• SERIRQ (12), or • Mouse Emulation Interrupt	
IRQ13	• P-Unit	
IRQ14	• IRQ14 from ISA IDE Interrupt	
IRQ15	• SERIRQ (15), or • IRQ15 from ISA IDE Interrupt	
IRQ16	• PIRQA	
IRQ17	• PIRQB	
IRQ18	• PIRQC	
IRQ19	• PIRQD	
IRQ20	• PIRQE, or • from SCI (based on ACTL.SCIS and PM1_CNT.SCI_EN registers)	3, 4
IRQ21	• PIRQF, or • from SCI (based on ACTL.SCIS and PM1_CNT.SCI_EN registers)	3, 4
IRQ22	• PIRQG, or • from SCI (based on ACTL.SCIS and PM1_CNT.SCI_EN registers)	3, 4
IRQ23	• PIRQH, or • from SCI (based on ACTL.SCIS and PM1_CNT.SCI_EN registers)	3, 4

Notes:
1. GCFG.LRE is the Legacy Rout Enable (LRE) bit 1 of the HPET General Configuration (GCFG) Register located at the memory-space address FED0_0010h.
2. When GCFG.LRE is set, the HPET T0C.IR and T1C.IR bits have no impact for timers 0 and 1. T0C and T1C are located at the memory-space addresses FED0_0100h and FED0_0120h, respectively.
3. When GCFG.LRE is cleared, each of the three HPET timers has its own routing control. The interrupts can be routed to various interrupts in the I/O APIC. T0C.IRC, T1C.IRC, and T2C.IRC indicate which interrupts are valid options for routing. If a timer is set for edge-triggered mode, the timers should not be shared with any other interrupts.
4. HPET Timer 2 is routed to the APIC as per the routing in the HPET T2C register located at the memory-space addresses FED0_0140h.
5. For routing of SCI, see Table 6-2.

The I/O APIC has a Redirection Table (RT) with an entry for each interrupt source. Each RT entry is individually programmed for trigger mode (edge or level), vector number, and destination processor(s). The interrupt is reported to the appropriate local APIC(s).

For more information about the I/O APIC, see Chapter 30, "I/O Advanced APIC (I/O APIC)."

6.7 8259 PIC Input Mapping

The interrupts that can be routed to the input of the integrated 8259 PIC are shown in Table 6-4. The 8259 PIC registers are described in Chapter 29, "8259 Programmable Interrupt Controller (PIC)."

Table 6-4. 8259 PIC Input Mapping

I/O PIC Input	Master or Slave 8259 PIC Input	Interrupts Routed to This PIC Input	Note
IRQ0	Master IRQ0	HPET Timer 0 (if GCFG.LRE is set) 8254 Timer (if GCFG.LRE is not set)	1
IRQ1	Master IRQ1	SERIRQ (1), or	
IRQ2	Master IRQ2	INTR output of the Slave 8259 PIC	1, 2
IRQ3	Master IRQ3	SERIRQ (3), or UART COM2, or PIRQx	3
IRQ4	Master IRQ4	SERIRQ (4), or UART COM1, or PIRQx	3
IRQ5	Master IRQ5	SERIRQ (5), or GPIO, or PIRQx	3
IRQ6	Master IRQ6	SERIRQ (6), or PIRQx	3
IRQ7	Master IRQ7	SERIRQ (7), or PIRQx	3
IRQ8	Slave IRQ0	HPET Timer 1 (if GCFG.LRE is set) RTC (if GCFG.LRE is not set)	1
IRQ9	Slave IRQ1	SERIRQ (9), or PIRQx, or from SCI (based on ACTL.SCIS and PM1_CNT.SCI_EN registers)	3
IRQ10	Slave IRQ2	SERIRQ (10), or PIRQx, or from SCI (based on ACTL.SCIS and PM1_CNT.SCI_EN registers)	3
IRQ11	Slave IRQ3	SERIRQ (11), or HPET Timer 2, or PIRQx, or from SCI (based on ACTL.SCIS and PM1_CNT.SCI_EN registers)	3
IRQ12	Slave IRQ4	SERIRQ (12), or PIRQx	3
IRQ13	Slave IRQ5	The interrupt (floating point error) is not supported.	
IRQ14	Slave IRQ6	SERIRQ (14), or IRQ15 from ISA IDE Interrupt, or PIRQx	3
IRQ15	Slave IRQ7	SERIRQ (15), or PIRQx	3

Notes:
1. Interrupts can individually be programmed to be edge or level, except for IRQ0, IRQ2, IRQ8#.
2. The slave 8259 controller is cascaded onto the master 8259 controller through the master controller interrupt input IRQ2.
3. For routing of the PIRQA through PIRQH Interrupts, see Table 6-1.

6.8 Device Interrupt-Generating Capabilities

Integrated devices that generate interrupts are shown in Table 6-5 and Table 6-6. The device that can generate a particular type of interrupt is marked with an "X." Interrupt-generating details of each device is located in its functional description chapter in this volume of the Datasheet.

Table 6-5. Device Interrupt Generating Capabilities

Device	INTx	MSI	MSI-X
Root Complex Event Collector (RCEC)	X	X	
Root Ports	X	X	X
SATA2 and SATA3	X	X	
USB 2.0	X		X
GbE	X	X	X
SMBus	X	X	
iLB Legacy	Refer to Chapter 21, "Intel Legacy Block (iLB) Devices"	Refer to Chapter 21, "Intel Legacy Block (iLB) Devices"	Refer to Chapter 21, "Intel Legacy Block (iLB) Devices"
UART	X		
Power Management			

Table 6-6. Device SCI, NMI, and SMI Generating Capabilities

Device	SCI	NMI	SMI
Root Complex Event Collector (RCEC)		X	X
Root Ports	X		X
SATA2 and SATA3			
USB 2.0			X
GbE			
SMBus			
iLB Legacy Refer to Chapter 21, "Intel Legacy Block (iLB) Devices"		X	X
UART			
Power Management	X		X

For some events, either an SCI or an SMI is generated. This is determined by the SCI Enable (SCI_EN) bit of the Power Management 1 Control (PM1_CNT) register. PM1_CNT is a 32-bit register located in the I/O space at ACPI_BASE_ADDRESS (ABASE) + 4.

Whether an SMI is generated, an SMI is controlled by the Global SMI Enable (GBL_SMI_EN) bit of the SMI Control and Enable (SMI_EN) register. SMI_EN is a 32-bit register located in the I/O space at ACPI_BASE_ADDRESS (ABASE) + 30h. The 32-bit SMI Status (SMI_STS) register is located in the I/O space at ACPI_BASE_ADDRESS (ABASE) + 34h.

§ §

7 SoC Reset and Power Supply Sequences

This chapter describes the platform board and SoC power-management signal interchange, SoC power-source sequencing requirements, and reset signaling for the various power states. The information is meant for platform-board designers. The SoC only supports the G3 (Mechanical Off) with or without an RTC coin-cell battery, S5 (Soft Off), and S0 (Fully On) states. The ACPI Sleep States (S1, S2, S3, S4) are not supported.

Ramp sequence for some SUS rails has been revised. Refer to Platform Design Guide for new recommendation.

7.1 Power Up from G3 State (Mechanical Off)

7.1.1 While in the G3 State

If the platform board provides a functional coin-cell battery for the SoC integrated Real Time Clock (RTC), the RTC power well of the SoC is functional during the G3 state. The voltage is supplied to the SoC by VCCRTC_3P3 (pin AG42).

When a coin-cell battery of sufficient voltage is inserted while in the G3 state, the platform board signals the SoC that the RTC power well voltage has been valid for a sufficient period of time for the SoC to clear its RTC registers. It does this by deasserting the RTC well RTEST_B and SRTCRST_B. See Figure 7-1 and Table 7-1. When the SRTCRST_B signal is deasserted, it indicates the end of RTC reset. When deasserted, RTEST_B signal indicates the RTC battery is producing a valid voltage. When a logic low, RTEST_B indicates a weak or missing RTC battery. The SoC makes the state of this RTEST_B signal accessible to software to detect a weak or missing RTC battery.

If the platform board does not have a functioning coin-cell battery, the SoC RTC power well ramps up the same time as V3P3A during the standby power-up sequence described in the following subsection. What occurs before that time is shown in Figure 7-2 and Table 7-1.

When no power other than the coin-cell battery is supplied to the SoC and when the active-low RTEST_B and SRTCRST_B SoC input signals are both logic-high levels (deasserted), it indicates that the SoC is provided what it needs to keep the RTC circuitry functioning during the G3 state. Unless they were set to their default values by an active-low signal on RTC well RTEST_B or SRTCRST_B, register bits located in the RTC well that were set/cleared while the SoC was previously in the S0 or S5 state are preserved by the SoC. These preserved register bits affect how the SoC reacts to future power events and its future power-state transitions.

7.1.2 Powering-Up for the First Time

The next subsection begins to describe the sequence for the case where the RTC-well register bits are at their default values (after the assertion of the RTEST_B SoC input signal by the platform board). This is refer to as “powering-up for the first time.” The SoC power-management mechanism has the capability of determining and remembering the first power-up situation.

Note: If the platform board does not have an RTC coin battery or if the RTC battery voltage is not valid, the SoC is always in a “powering-up for the first time” situation as it transitions from the G3 state to the S5 state.

7.1.3 SUS Power Well Power-Up Sequence From the G3 State

Refer to Figure 7-1, Figure 7-2, and Table 7-1.

Some of the pin-based straps (hard straps) values are sampled by the SoC when the active-low RSMRST_B signal is deasserted. These "SUS Power OK" hard strap values must be valid for at least 400 ns after RSMRST_B signal is deasserted. Hard straps are described in Section 16.2, "Pin-Based (Hard) Straps" on page 357. Important to the SoC power-up sequence is the sampling of the SoC pin Y65 which is the "After G3 Enable" (AG3E) hard strap. Its effect is mentioned later.

Figure 7-1. Power-Up SUS Power Well Voltages to S5 State (with RTC Battery)

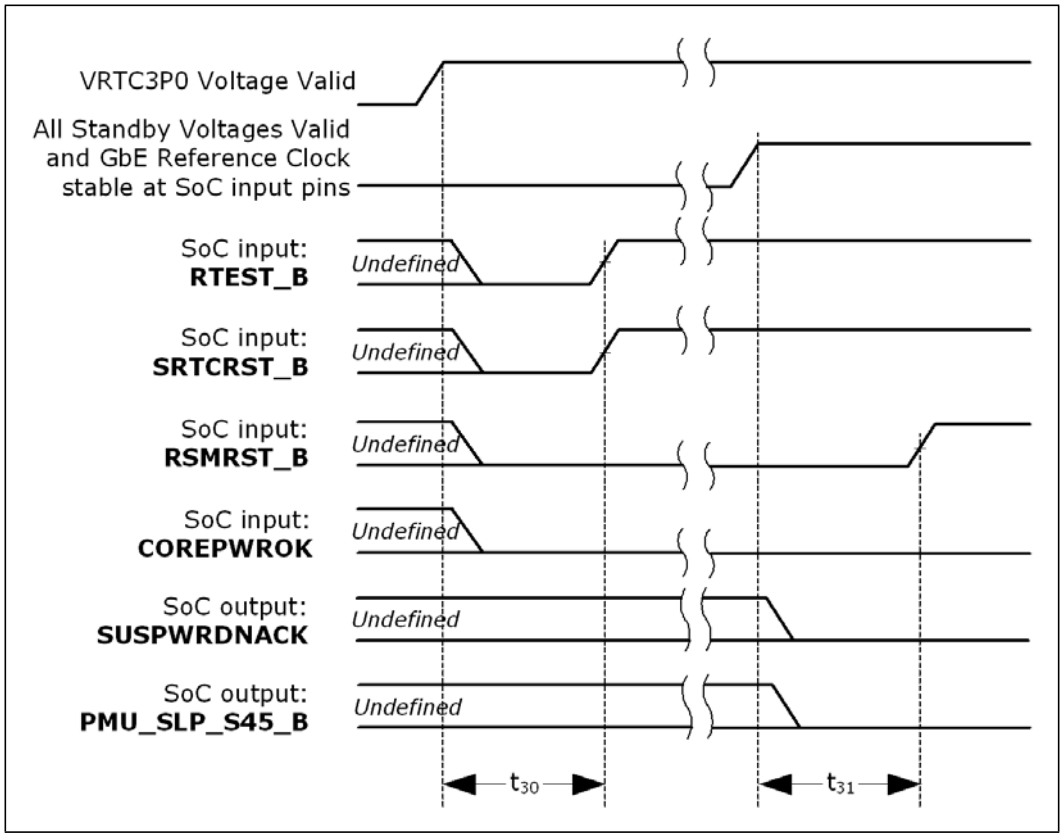


Figure 7-2. Power-Up SUS Power Well Voltages to S5 State (when no RTC Battery Voltage)

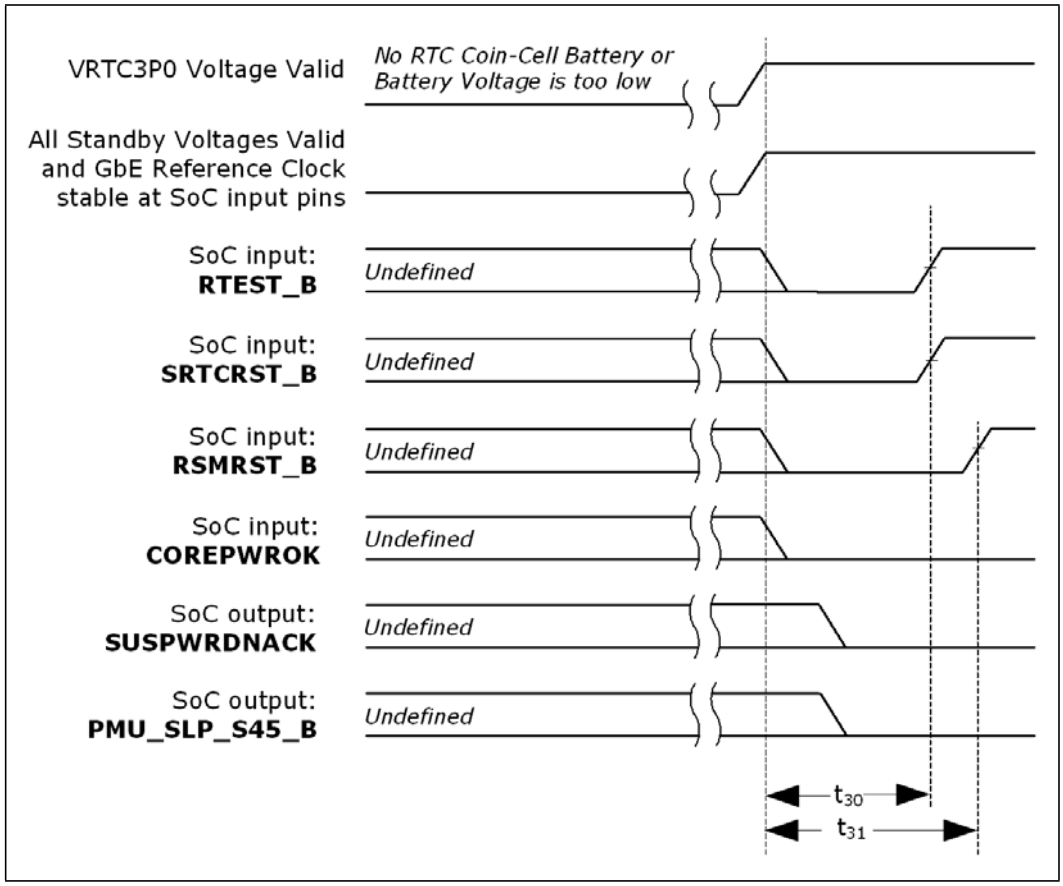


Table 7-1. Power-Up SUS Power Well Voltages to S5 State

Sym	Parameter	Min	Max	Units	Note	Fig
t30	RTEST_B deasserted by platform board after VRTC3P0 voltage valid at SoC pins. SRTCRST_B deasserted by platform board after VRTC3P0 voltage valid at SoC pins.	9	-	ms	-	7-1 7-2
t31	RSMRST_B deasserted by platform board after all standby voltages are valid and GbE Reference Clock stable at SoC input pins	10	-	ms	1	7-1 7-2
-	"SUS Power OK" hard strap value hold time after RSMRST_B de-asserted by the platform board	400	-	ns	1	7-1 7-2

Note:
1. GbE Reference Clock input-pin signals are GBE_REFCLK[P, N] (differential input). If the platform design does not need the integrated Ethernet Controller to be enabled during the S5 state, the timing parameter duration is measured only when all standby voltages are valid.

The platform board initiates the power-up sequence by providing the SUS power well (also called "Always On" and designated by the suffix "A") supply voltages to the SoC. To ensure proper SoC operation and avoid damaging the SoC, the standby voltage groups must be applied to the SoC in a particular order. The order is shown below.

During this sequence, the platform asserts the active-low RSMRST_B signal, powered by the RTC rail, to the SoC. Also, the platform board must drive the active-high SoC input COREPWROK low indicating that the Core power well pins do not have their valid voltage levels. If the platform board does not have a functioning coin-cell battery, these signals will not be detected by the SoC until the V3P3A voltage is provided to the SoC.

Before advancing to the next step within the sequence, the voltage supplied to the SoC must be regulated and at a valid voltage level. Voltage levels are measured at the SoC pins/balls. See Table 34-3, "Voltage Supply Requirements Under Normal Operating Conditions" on page 690 for valid voltage levels for each voltage group. Here is the sequence:

1. V1P8A voltage is provided to all V1P8A voltage-group pins of the SoC.
2. V1P0A voltage is provided to all V1P0A voltage-group pins the of SoC.
   — It is permissible for V1P8A and V1P0A to be powered-up at the same time, but it is best to stagger their ramp-up as indicated here or as V1P8A first, then V1P0A.
3. Wait for both V1P8A and V1P0A to be regulated and at valid voltage levels.
4. V3P3A voltage is provided to all V3P3A voltage-group pins of the SoC.

Once V3P3A is valid for a period of time, the platform indicates to the SoC that all standby voltages are up and valid. It does this by deasserting the RSMRST_B signal. See Figure 7-1, Figure 7-2, and Table 7-1.

Once V3P3A is valid and the RSMRST_B signal is deasserted, the active-low SoC output signal PMU_SLP_S45_B becomes valid and asserted (logic-low level) indicating that the SoC is in the S5 (Soft Off) state. Whenever a logic high level, the RSMRST_B signal indicates to the SoC that the platform board is providing the SUS power wells with valid voltage (a.k.a, standby voltage).

The SoC is now in the S5 (Soft Off) state. Only the SUS power well and RTC power well are active within the SoC.

7.1.4 Core Power-Up Sequence

Refer to Figure 7-3 and Table 7-2. At this point the SoC either:

- If "Powering-Up for the First Time" the After G3 Enable (AG3E) hard pin strap (SoC pin Y65, see Table 16-1, "Hard Pin Straps" on page 357) determines the next sequence of events. This hard pin strap was sensed when the standby power-up sequence completed (Section 7.1.3, "SUS Power Well Power-Up Sequence From the G3 State" on page 126):
  - SoC pin Y65 = logic low: The SoC exits S5 and proceeds with sequence to ultimately get to the S0 state.
  - SoC pin Y65 = logic high: The SoC remains in S5 until a Wake Event occurs (e.g., PMU_PWRBTN_B active for less than 4 seconds). When the event occurs, it proceeds with sequencing to ultimately get to the S0 state.

Note: If the platform board does not have an RTC coin battery or if the RTC battery voltage is not valid, the SoC is always in a "Powering-Up for the First Time" situation as it transitions from the G3 to S5 state.

- If not "Powering-Up for the First Time" the AFTERG3_EN (AG3E) register bit, bit 0 of the GEN_PMCON1 register, determines the next sequence of events:
  - AG3E register = 0: The SoC exits S5 and proceeds with sequence to ultimately get to the S0 state.
  - AG3E register = 1: The SoC remains in S5 until a Wake Event occurs (e.g., PMU_PWRBTN_B active for less than 4 seconds). When the event occurs, proceed with sequencing to ultimately get to the S0 state.

Note: BIOS can write the AG3E register bit for use on subsequent G3-to-S5 transitions. Only the "Powering-Up for the First Time" situation uses the AG3E hard pin strap for this purpose.

The Wake Events are listed in Section 19.3.1, "Reset Behavior" on page 448.

When the SoC proceeds, it does so by exiting the S5 state and deasserting the active-low SoC output signal PMU_SLP_S45_B. At this time, the SoC is ready to begin receiving the DDR3 and core-well voltages and remaining reference clocks from the platform board.

The platform board may now begin to apply power to the DDR3 circuitry and to the core power well of the SoC. These voltage groups are considered "switched" voltages in that they are not always on like the standby voltage groups. Some of the voltage group names have the suffix "S."

During this power-up sequence, the platform board continues to deassert the active-high SoC COREPWROK input signal indicating that the Core power-well pins do not have their valid voltage levels, and the reference clocks are not valid and stable.

Note: All voltage-supply sequencing requirements given in this chapter are specified at the SoC pins/balls.

Figure 7-3. S5 State to S0 State Sequence - Not Cold Reset

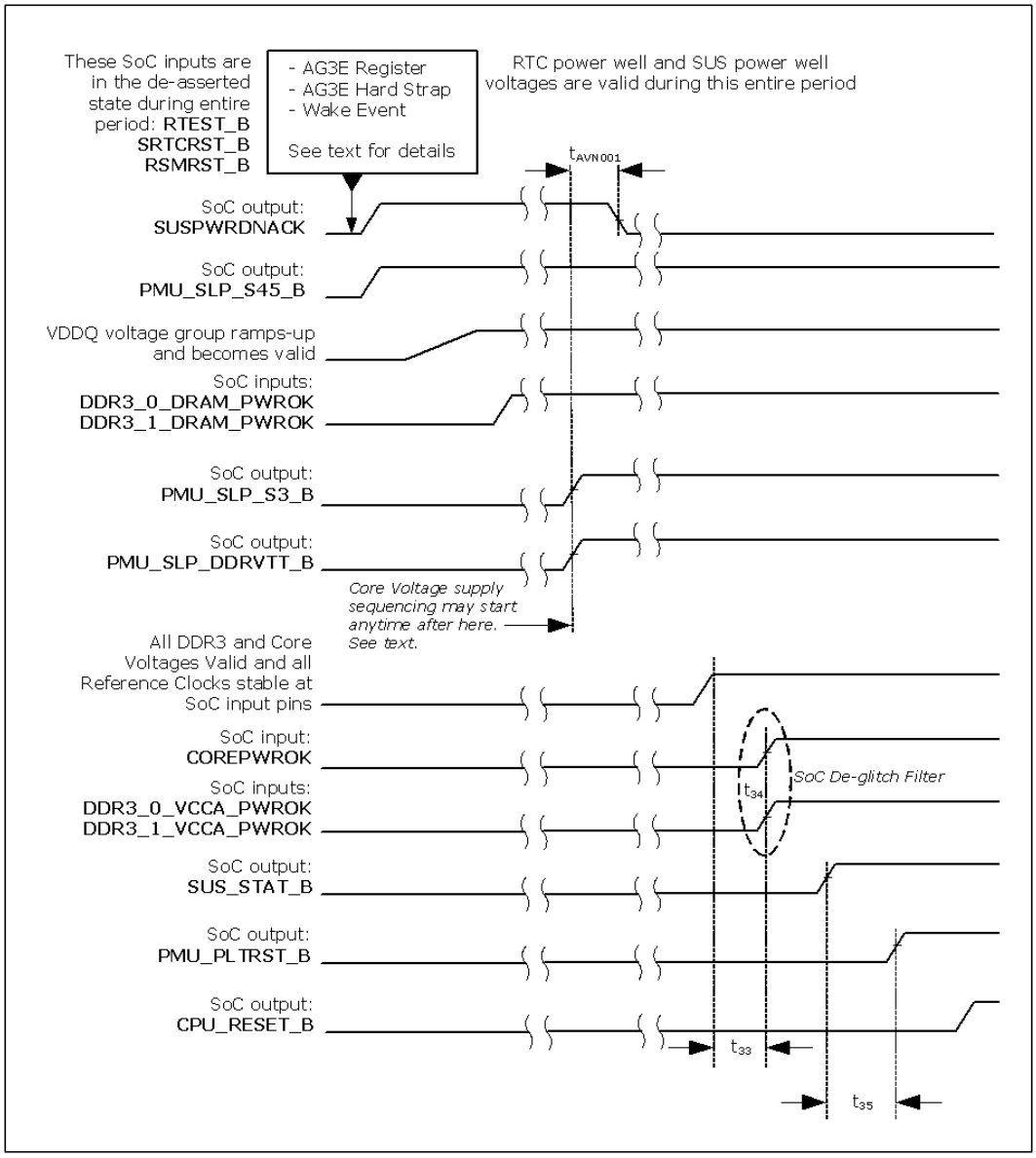


Table 7-2. S5 State to the S0 State Sequence - Not Cold Reset

Sym	Parameter	Min	Max	Units	Note	Fig
t33	COREPWROK asserted after all DDR3 and Core voltages are valid and all Reference Clocks stable at SoC input pins	10	-	ms	1, 2, 3	7-3
t34	COREPWROK, DDR3_0_VCCA_PWROK, and DDR3_1_VCCA_PWROK active logic-level duration required to be sensed as valid by the SoC.	1	-	ms		7-3
-	"COREPWROK" hard strap value hold time after COREPWROK asserted by platform board	400	-	ns	1	7-3
t35	PMU_PLTRST_B de-asserted after SUS_STAT_B de-asserted	60	100	µs	4	7-3
tAVN001	SUSPWRDNACK de-asserts after PMU_SLP_DDRVTT_B de-asserts	-	200	ns		7-3

Notes:
1. Some pin-based hard straps are sampled before COREPWROK is asserted. The SoC latches these strap values when COREPWROK transitions to the asserted state.
2. Reference Clock input-pin signals are:
   HPLL_REF[P, N] (differential input)
   PCIE_REFCLK[P, N] (differential input)
   SATA_REFCLK[P, N] (differential input)
   SATA3_REFCLK[P, N] (differential input)
   GBE_REFCLK[P, N] (differential input)
   USB_REFCLK[P, N] (differential input)
3. When the SoC output signal PMU_PLTRST_B is used by the platform board design to provide PCI Express* components or add-in adapter cards the PCI Express* Fundamental Reset signal called PERST#, refer to Section 2.6.2 of the *PCI Express Card Electromechanical Specification*, Revision 2.0. It specifies special Power Sequencing and Reset Signal Timings that supersede the t33 parameter in this table.
4. The Min parameter allows satisfying the 30-µs minimum requirement show in Figure 9: Timing for Entering and Exiting the Power Down of the *Intel Low Pin Count (LPC) Interface Specification, Revision 1.1*.

The platform board may now begin to apply power to the DDR3 circuitry and to the Core power well of the SoC. These voltage groups are considered "Switched" voltages in that they are not always on like the standby voltage groups. Some of the voltage group names have the suffix "S."

During this power-up sequence, the platform board continues to drive the active-high SoC input COREPWROK low indicating that the Core power well pins do not have their valid voltage levels and the reference clocks are not valid and stable.

Note: All voltage-supply sequencing requirements are given as measured at the SoC pins/ balls.

See Table 34-3, "Voltage Supply Requirements Under Normal Operating Conditions" on page 690 for the valid voltage levels for each voltage group. The power-up sequence begins with VDDQ:

1. VDDQ (VDDQA/VDDQB)
   - Using VDDQA (Channel 0) and VDDQB (Channel 1) instead of a single VDDQ voltage source is based on the DDR3 memory component topology of the platform board.
   - Using VDDQA and VDDQB instead of a single VDDQ is based on the DIMM topology. When the DIMMs are on either side of the SoC, VDDQA and VDDQB are used for Channel 0 and Channel 1 respectively.
   - When VDDQ is valid, the platform asserts the DDR3_0_DRAM_PWROK and DDR3_1_DRAM_PWROK. The SoC receivers for these signals are powered by VDDQ.
   - The SoC is now in the S3 state. Because the SoC does not support S3, the SoC does not remain in S3.
   - When the SoC is ready to exit the S3 state and advance to the S0 state, it deasserts the output signals PMU_SLP_S3_B and PMU_SLP_DDRVTT_B. The platform board design may use the PMU_SLP_DDRVTT_B signal to provide power to the SDRAM components.
2. This step is optional: Wait for PMU_SLP_S3_B and PMU_SLP_DDRVTT_B output signals to de-assert.
3. VNN and VCC may begin to ramp-up together.
4. Once VNN and VCC voltages are valid and stable at the SoC pins, VCCSRAM may begin to ramp-up at the SoC pins no later than 5 ms. Designers should make this delay as short as possible.
5. As VCC begins to ramp-up, V1P35S may begin to ramp-up.
6. Once VNN voltage is valid and stable, V1P0S may begin to ramp-up.
7. Once V1P0S begins to ramp-up, V1P8S may begin to ramp-up.
8. VNN, VCC, VCCSRAM, V1P35S, V1P0S, and V1P8S are valid and stable.
9. V3P3S may begin to ramp-up.

See Figure 7-3 and Table 7-2. Once the platform board has all of the DDR3 and core power well voltage supplies at their valid voltages, and all of the reference clocks are stable at the SoC input pins, it asserts the COREPWROK signal to the SoC. At the same time, the platform also asserts the DDR3_0_VCCA_PWROK and DDR3_1_VCCA_PWROK SoC input signals.

Some of the pin-based straps (hard straps) values are sampled by the SoC when the platform board asserts the COREPWROK signal to the SoC. These hard strap values must be valid for at least 400 ns after the COREPWROK signal is asserted. Hard straps are described in Section 16.2, "Pin-Based (Hard) Straps" on page 357.

The SoC then deasserts SUS_STAT_B and platform reset (PMU_PLTRST_B).

The platform board and the SoC are now ready to begin functioning in the S0 state. The SoC internal reset for the core CPU used for the BIOS is completed, and the BIOS instruction fetching begins from the Flash memory. This core reset is also used for the SoC output signal, CPU_RESET_B, which the platform board provides to the In-Target Probe (ITP) connector if part of the board design. It is used only for debug purposes.

7.2 Reset Sequences and Power-Down Sequences

The SoC remains in the S0 (Working, or Fully-On) until some event causes it to:

1. Perform a Cold Reset, or
2. Perform a Warm Reset, or
3. In an orderly fashion, transition to, and remain in the S5 (Soft Off) state, or
4. In a quick fashion, transition to, and remain in the S5 (Soft Off) state, or
5. In an orderly fashion, transition to and remain in the G3 (Mechanical Off) state.

In some cases, before the SoC exits from the S0 state, the event causing the exit is retained by the SoC and preserved by the SUS well voltage or by the RTC battery voltage. The retained information is used by the SoC to transition back to S0 in the appropriate manner.

7.2.1 Cold Reset Sequence

From a platform board perspective, the SoC initiates a Cold Reset sequence while in the S0 (Working) state. The situations when this occurs are shown as Reset Types 2 and 4 in Table 19-4, "SoC Reset Sources" on page 448 in Section 19.3.1, "Reset Behavior" on page 448.

During the entire Cold Reset sequence, the platform maintains valid voltage levels for the SoC SUS well (Standby power). The following 20-step sequence is performed.

Refer to Figure 7-4 and Table 7-3 for sequence steps 1 through 9:

1. The SoC begins the sequence by asserting the active-low SUS_STAT_B output signal to the platform board.
2. The SoC initiates a Platform Reset by asserting the active-low PMU_PLTRST_B output signal to the platform board.
3. The SoC asserts the active-low PMU_SLP_S3_B and PMU_SLP_DDRVTT_B output signals to the platform board.
   — This indicates that the SoC has entered the S3 state. This occurs even though the SoC does not support the Suspend to RAM sleep state (S3).
4. The platform board responds by deasserting the active-high COREPWROK, DDR3_0_VCCA_PWROK, and DDR3_1_VCCA_PWROK input signals of the SoC.
5. The platform board removes the SoC Core well and VDDQ voltages in the following sequence:
   a. Power to the board SDRAM components.
   b. V3P3S
   c. V1P8S
   d. VCCSRAM
   — Optionally, VCC may power-down the same time as VNN shown in step g below.
   e. V1P0S
   f. V1P35S
   — V1P35S must power down before VNN or both rails (V1P35S and VNN) can be powered down simultaneously
   g. VNN
6. The SoC now enters the S5 (Soft Off) state as asserts the active-low PMU_SLP_S45_B output signal to the platform board.

7. The platform board responds by deasserting the active-high, DDR3_0_DRAM_PWROK, and DDR3_1_DRAM_PWROK input signals of the SoC.
8. The platform board removes the VDDQ (VDDQA and VDDQB) voltages from the SoC.
9. The SoC is now ready to exit the S5 (Soft Off) state.

Refer to Figure 7-5 and Table 7-4 for steps 10 through 20:

10. The SoC now exits the S5 (Soft Off) state by deasserting the active-low PMU_SLP_S45_B output signal to the platform board
11. VDDQ (VDDQA/VDDQB)
    - When VDDQ is valid, the platform asserts the DDR3_0_DRAM_PWROK and DDR3_1_DRAM_PWROK. The SoC receivers for these signals are powered by VDDQ.
    - The SoC is now in the S3 state. Because the SoC does not support S3, the SoC does not remain in S3.
    - When the SoC is ready to exit the S3 state and advance to the S0 state, it deasserts the output signals PMU_SLP_S3_B and PMU_SLP_DDRVTT_B. The platform board design may use the PMU_SLP_DDRVTT_B signal to provide power to the SDRAM components.
12. This step is optional: Wait for PMU_SLP_S3_B and PMU_SLP_DDRVTT_B output signals to de-assert.
13. VNN and VCC may begin to ramp-up together.
14. Once VNN and VCC voltages are valid and stable at the SoC pins, VCCSRAM may begin to ramp-up at the SoC pins no later than 5 ms. Designers should make this delay as short as possible.
15. As VCC begins to ramp-up, V1P35S may begin to ramp-up.
16. Once VNN voltage is valid and stable, V1P0S may begin to ramp-up.
17. Once V1P0S begins to ramp-up, V1P8S may begin to ramp-up.
18. VNN, VCC, VCCSRAM, V1P35S, V1P0S, and V1P8S are valid and stable.
19. V3P3S may begin to ramp-up.
20. Once the platform board has all of the DDR3 and core power well voltage supplies at their valid voltages, and all of the reference clocks are stable at the SoC input pins, it asserts the COREPWROK signal to the SoC. At the same time, the platform also asserts the DDR3_0_VCCA_PWROK and DDR3_1_VCCA_PWROK SoC input signals.
21. The SoC then deasserts SUS_STAT_B and platform reset (PMU_PLTRST_B).

The platform board and the SoC are now ready to function in the S0 state. The SoC internal reset for the core CPU used for the BIOS is completed, and the BIOS instruction fetching begins from the Flash memory. This core reset is also used for the SoC output signal, CPU_RESET_B, which the platform board provides to the In-Target Probe (ITP) connector if part of the board design. It is used only for debug purposes.

Figure 7-4. S0 State to S5 State Sequence

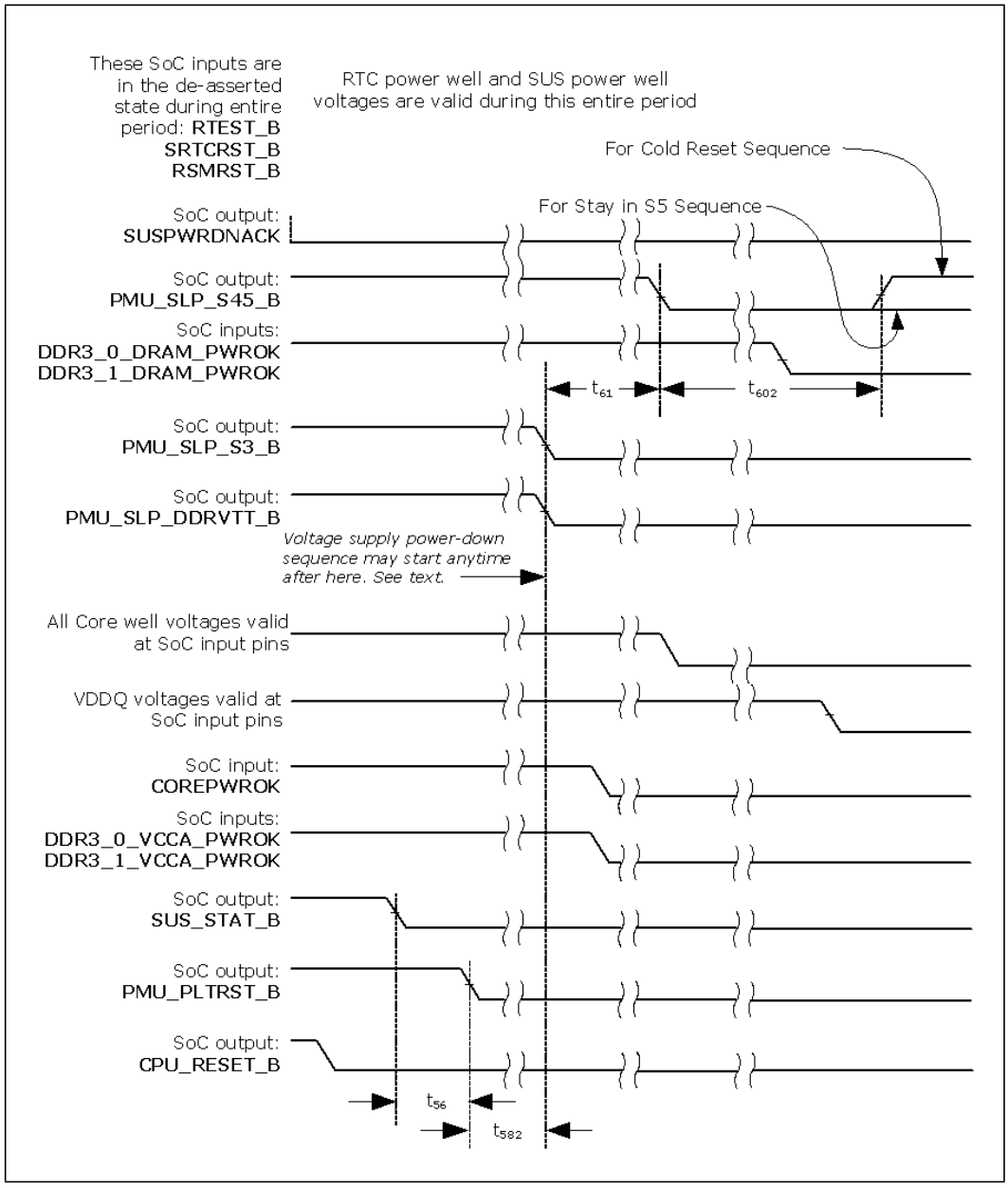


Table 7-3. S0 State to S5 State Sequence

Sym	Parameter	Min	Max	Units	Note	Fig
t56	PMU_PLTRST_B asserted by SoC after SUS_STAT_B asserted by SoC	60	-	µs	1	7-4
t582	PMU_SLP_DDRVTT_B and PMU_SLP_S3_B asserted by SoC after PMU_PLTRST_B asserted by SoC	31	-	µs		7-4
t61	PMU_SLP_S45_B asserted by SoC after PMU_SLP_S3_B asserted by SoC	30	-	µs	2	7-4
t602	PMU_SLP_S45_B deasserted by SoC after PMU_SLP_S45_B asserted by SoC.	4	5	sec		7-4

Notes:
1. The minimum parameter allows satisfying the 30-µs minimum requirement show in Figure 9: Timing for Entering and Exiting the Power Down of the *Intel Low Pin Count (LPC) Interface Specification, Revision 1.1*.
2. The SoC can be configured by software to stretch the duration of PMU_SLP_S3_B.

Figure 7-5. S5 State to S0 State Sequence - Cold Reset

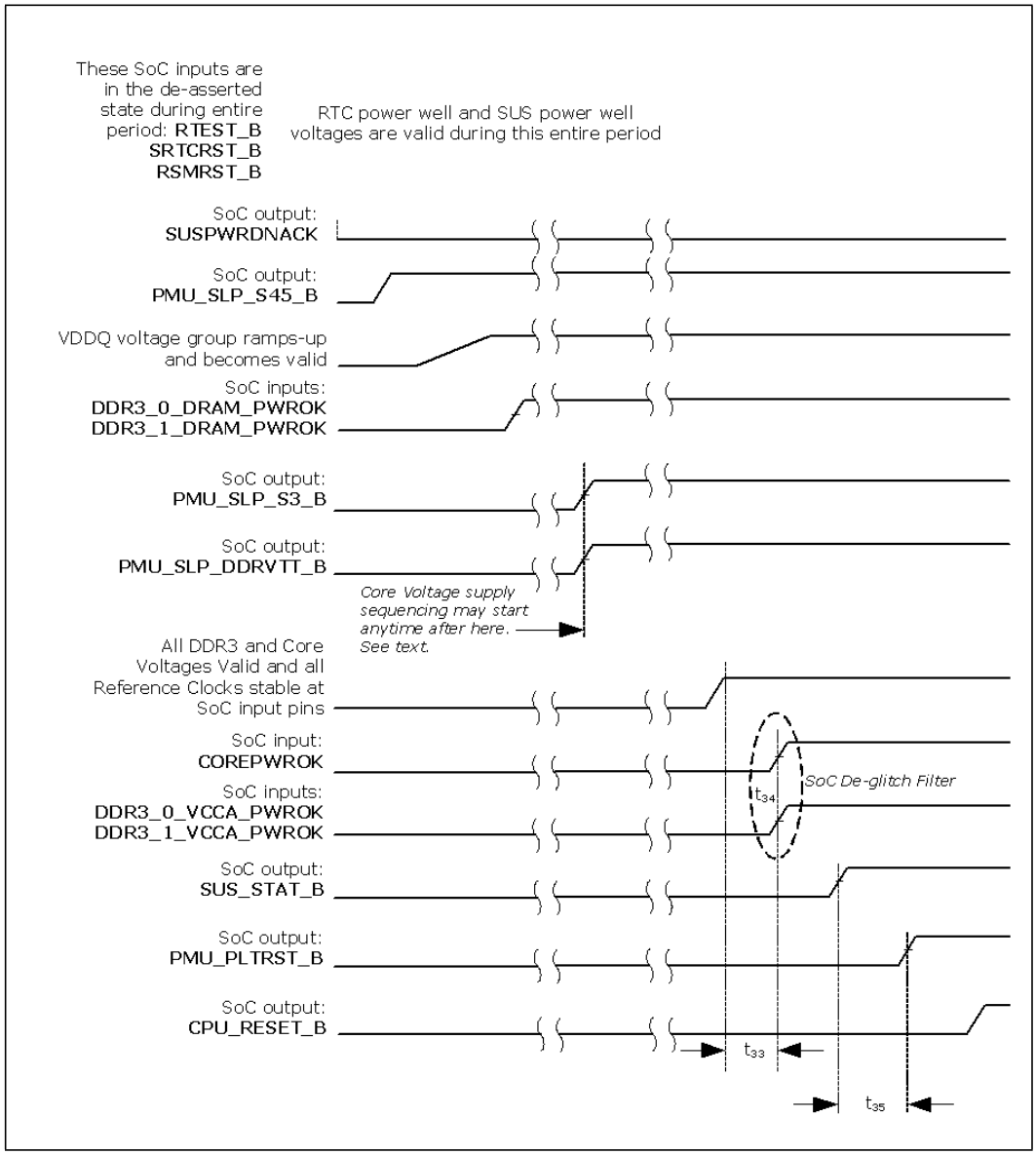


Table 7-4. S5 State to S0 State Sequence - Cold Reset

Sym	Parameter	Min	Max	Units	Note	Fig
t33	COREPWROK asserted after all DDR3 and Core voltages are valid and all Reference Clocks stable at SoC input pins	10	-	ms	1, 2, 3	7-5
t34	COREPWROK, DDR3_0_VCCA_PWROK, and DDR3_1_VCCA_PWROK active logic-level duration required to be sensed as valid by the SoC.	1	-	ms		7-5
t35	PMU_PLTRST_B de-asserted after SUS_STAT_B de-asserted	60	100	µs	4	7-5

Notes:
1. Some pin-based hard straps are sampled before COREPWROK is asserted. The SoC latches these strap values when COREPWROK transitions to the asserted state.
2. Reference Clock input-pin signals are:
   HPLL_REF[P, N] (differential input)
   PCIE_REFCLK[P, N] (differential input)
   SATA_REFCLK[P, N] (differential input)
   SATA3_REFCLK[P, N] (differential input)
   GBE_REFCLK[P, N] (differential input)
   USB_REFCLK[P, N] (differential input)
3. When the SoC output signal PMU_PLTRST_B is used by the platform board design to provide PCI Express* components or add-in adapter cards the PCI Express* Fundamental Reset signal called PERST#, refer to Section 2.6.2 of the *PCI Express Card Electromechanical Specification*, Revision 2.0. It specifies special Power Sequencing and Reset Signal Timings that supersede the t33 parameter in this table.
4. The Min parameter allows satisfying the 30-µs minimum requirement show in Figure 9: Timing for Entering and Exiting the Power Down of the *Intel Low Pin Count (LPC) Interface Specification, Revision 1.1*.

7.2.1.1 SUSPWRDNACK

This SoC output signal allows the platform board, as an option, to power-down the SoC SUS well during the S5 state. The SoC does not support this situation and so the SUSPWRDNACK output signal is always inactive during a Cold Reset sequence.

The SoC does assert the SUSPWRDNACK signal for a brief period after the SUS well voltages are powered-on. See Figure 7-3.

7.2.2 Warm Reset Sequence

From a platform board perspective, the SoC initiates a Warm Reset sequence while in the S0 (Working) state. The situations when this occurs are shown as Reset Type 1 in Section 19.3.1, "Reset Behavior" on page 448.

A Warm Reset sequence is the same as the Cold Reset except that the SoC SUS well, DDR3 power, and Core well power remain on during the entire sequence. The platform also provides valid reference clocks to the SoC during the entire Warm Reset sequence. Refer to Figure 7-6 and Table 7-5:

1. The SoC issues a Platform Reset (PMU_PLTRST_B).
2. The SoC remains in the S0 (Working) state.
3. The SoC ends the Platform Reset.
4. The SoC re-boots the BIOS and operating system.

Figure 7-6. Warm Reset Sequence

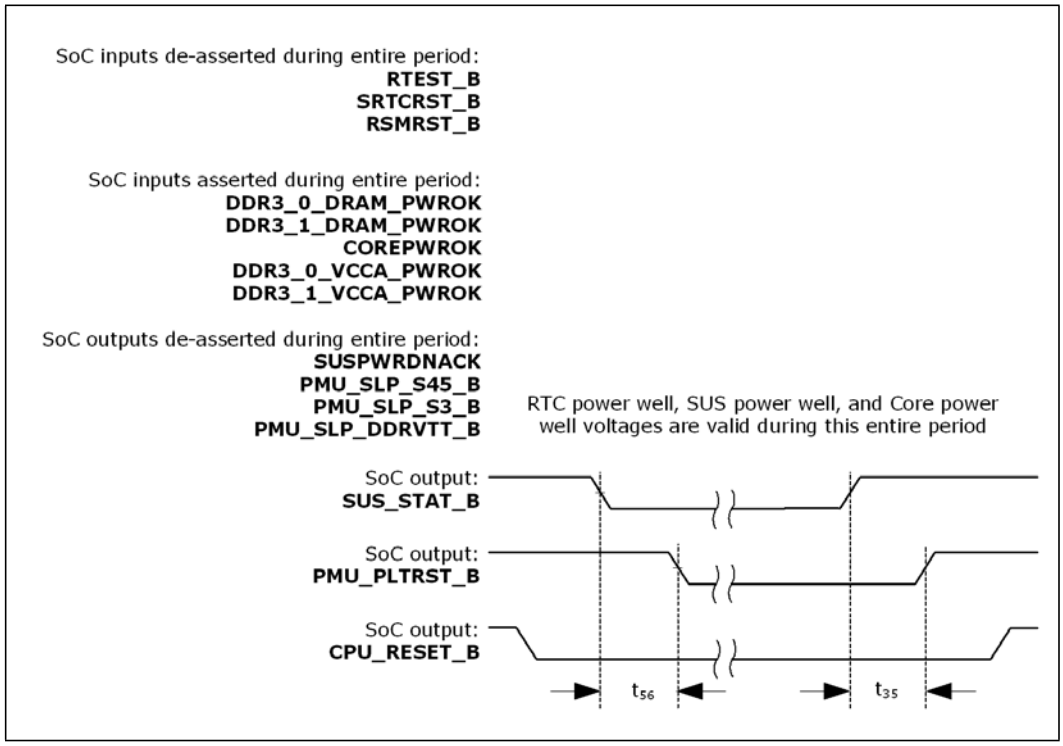


Table 7-5. Warm Reset Sequence

Sym	Parameter	Min	Max	Units	Note	Fig
t56	PMU_PLTRST_B asserted by SoC after SUS_STAT_B asserted by SoC	60	-	µs	1	7-6
t35	PMU_PLTRST_B de-asserted after SUS_STAT_B de-asserted	60	100	µs		7-6

Note:
1. The Min parameter allows satisfying the 30-µs minimum requirement show in Figure 9: Timing for Entering and Exiting the Power Down of the *Intel Low Pin Count (LPC) Interface Specification, Revision 1.1*.

7.2.2.1 SPD Reset Sequence

After power-up, the BIOS might detect the DDR3 DIMM/SODIMM SDRAM frequency through Serial Presence Detect (SPD) using the SMBus interface. In this case, the BIOS might initiate a Warm Reset to re-generate and re-lock the internal SoC memory-controller clocks. From a platform board perspective, the SPD Reset sequence is the same as the Warm Reset sequence.

7.2.3 Power-Down to S5 (Soft Off) and Stay There Sequence

The SoC can initiate a power-down to S5 sequence while in the S0 (Working) state. The situations when this occurs are shown as Reset Types 3 and 5 in Table 19-4, "SoC Reset Sources" on page 448 in Section 19.3.1, "Reset Behavior" on page 448.

Reset Type 3 is an orderly transition to S5. Reset Type 5 is a quick transition to S5 and usually caused by some kind of unexpected time out condition but actually appears like a Type-3 reset to the platform board.

The sequence from S0 to S5 is the same as sequence steps 1 through 8 shown Section 7.2.1, "Cold Reset Sequence" on page 133. Refer to Figure 7-4 and Table 7-3.

The SoC remains in the S5 state until a Wake Event occurs. Wake Event hardware is powered by the SUS well and RTC well power which remain powered-on during S5.

7.2.4 Events While Sleeping in S5 (Soft-Off) State

When sleeping in the S5 state, the SoC transitions to either the S0 (Fully-On) or the G3 (Mechanical-Off) state when certain configured Wake Events occur. S5 transitions to G3 are always initiated by the platform board via the RSMRST_B signal. The Wake Events are shown in Section 19.3.3, "Exiting the G2 (S5) Soft-Off Power State" on page 452.

While the SoC is in the S5 state, the active-low SoC output signal PMU_SLP_S45_B remains asserted.

7.2.4.1 S5 to S0 State

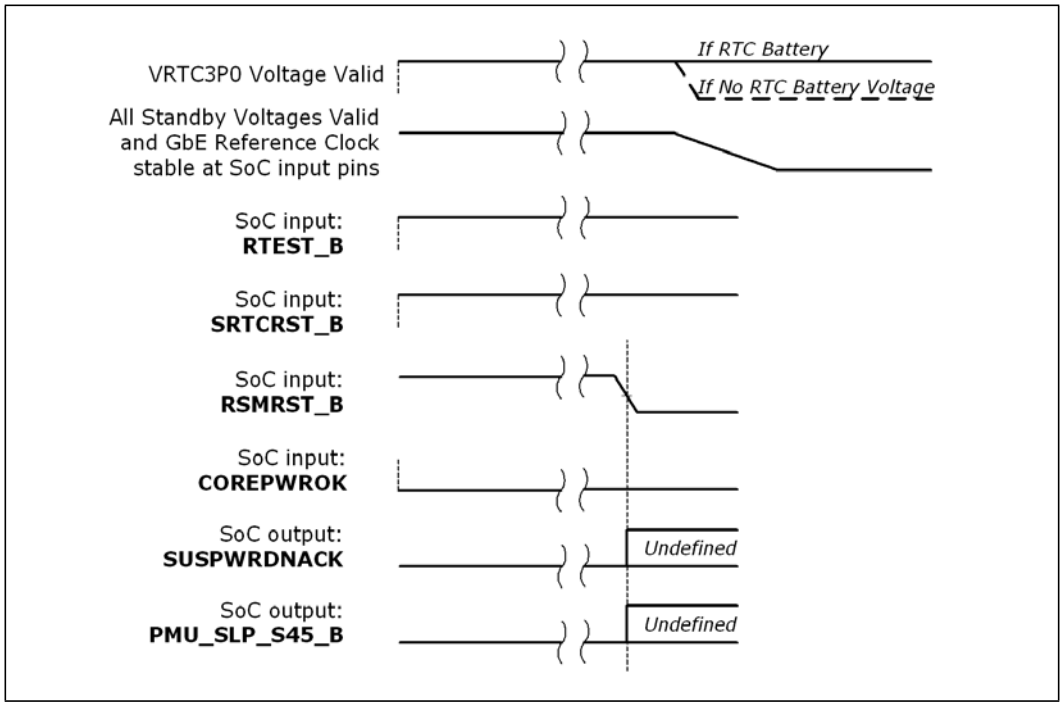
See Section 7.1.4, "Core Power-Up Sequence" on page 129 for sequences when powering up from the G3 state with an active After G3 Enable (AG3E) register bit or with the SoC AG3E hard strap active.

When a configured Wake Event occurs while the SoC is sleeping in the S5 state, the SoC exits the S5 state, and deasserts the active-low SoC output signal PMU_SLP_S45_B. This is an indication to the platform board to begin powering-up the DDR3 voltages and the SoC Core well voltages. See Figure 7-3 and Table 7-2 for the signal and power-rail sequences and timing parameters.

7.2.4.2 S5 to G3 State

While deasserted by the platform board, the active-low Resume Well Reset (RSMRST_B) signal indicates to the SoC that the platform board is supplying valid SUS well power to the SoC. If the platform board asserts RSMRST_B during the S5 state, the SoC enters the G3 (Mechanical Off) state and the platform board may power-down the SoC SUS well (standby) voltages. The power-down sequence of the SoC SUS well (standby) voltages is shown in Section 7.2.4.3, "SUS Well Power Down Sequence" on page 142.

Figure 7-7. S5 State to G3 State Sequence



7.2.4.3 SUS Well Power Down Sequence

In the S5 state, the DDR3 and the SoC Core well power is off. The platform board powers down the SoC SUS well (standby) voltages in the following manner:

Note: All voltage-supply sequencing requirements are given as measured at the SoC pins/balls.

a. V3P3A
b. V1P0A
c. V1P8A

— Optionally, the following SUS well power-down sequence may be used:

a. V3P3A
b. V1P8A
c. V1P0A

Note: It is permissible for V1P8A and V1P0A to be powered-up at the same time, but it is best to stagger their ramp-up as indicated here.

7.2.5 Power-Down from S0 to G3 (Mechanical Off) Sequence

During an orderly power-down from S0 to the G3 state, the SoC first enters the S3 state and then the S5 state as shown in Section 7.2.3, "Power-Down to S5 (Soft Off) and Stay There Sequence" on page 141. The sequence from S5 to G3 is shown in Section 7.2.4, "Events While Sleeping in S5 (Soft-Off) State" on page 141.

§ §

8 Thermal Management

8.1 Overview

The SoC implements configurable forms of thermal management for itself, memory, and the system. The architecture implements various proven methods of maintaining maximum performance while remaining within thermal specifications. Thermal Control Circuit (TCC) mechanisms are used to reduce power consumption when thermal device limits are exceeded and the system is notified of this condition via interrupts or thermal signaling pins.

Thermal management features include:

- Up to 15 Digital Thermal Sensors (DTS) per SoC
  - Three sensors per two-core module
  - Three uncore sensors
- Four (4) thermal interrupt triggers per sensor
  - Hot, critical, and two programmable thresholds
- PROCHOT_B, MEMHOT_B, and THERMTRIP_N
- Thermal Control Circuit (TCC) mechanisms
  - Bi-directional PROCHOT_B
  - Intel® Thermal Monitor 1
  - Intel® Thermal Monitor 2
- Thermal monitoring and actions managed by the SoC
- Closed-Loop Thermal Throttling (CLTT) pass-through
- Temperature provided by the BMC or other external circuitry via PECI over SMBus
- Memory thermal control
- Fan Speed Control (FSC) parameter ($T_{CONTROL}$)

Table 8-1. References

Reference	Revision	Date	Document Title
ACPI Specification	5.0	December 6, 2011	*Advanced Configuration and Power Interface Specification*, Revision 5.0

The functional description for PECI over the integrated SMBus is in Chapter 17, "SMBus 2.0 Unit 2 - PECI."

8.2 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The signal/pin name
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

See Chapter 31, "Signal Names and Descriptions" for details of these signals.

Table 8-2. Signals Mentioned in This Chapter

Signal Name	Direction and Type	Description
PROCHOT_B	I/O Open-Drain	**Processor Hot:** Active low. As an output, the signal is asserted if any SoC thermal sensor indicates the component is hot. As an input, the platform board External Circuitry (EC) chooses to drive this signal low to reduce SoC current consumption if the EC detects overheating.The platform board must ignore this SoC output signal while PMU_PLTRST_B (active-low SoC output) is asserted.
MEMHOT_B	Input	**Memory Hot:** Active-low input signal from the platform board indicating a memory-overheating condition. When this signal is active, the SoC performs memory throttling in an attempt to cool the memory devices.The platform board must ignore this SoC output signal while PMU_PLTRST_B (active-low SoC output) is asserted.
THERMTRIP_N	Output	**Catastrophic Thermal Trip:** Active-low output signal indicating the SoC has reached an operating temperature that may damage the component. See Section 8.5.7, "THERMTRIP_N Signal" on page 149 for additional details on THERMTRIP_N.
SMB_CLK2	I/O Open-Drain	**SMBus Clock (SMBCLK):** For PECI over SMBus interface with the BMC or other platform board External Circuitry. *This signal is muxed and is used by other functions.*
SMB_DATA2	I/O Open-Drain	**SMBus Data (SMBDAT):** For PECI over SMBus interface with the BMC or other platform board external circuitry. *This signal is muxed and is used by other functions.*

8.3 CPU Thermal Management Registers

Thermal management control and status registers are distributed between Model Specific Registers (MSR) in each core, Power Management Link (PMLink) configuration registers, and P-Unit registers. Further register descriptions which control thermal management features are found in the *Intel® Atom™ Processor C2000 Product Family BIOS Writer's Guide (BWG)*.

8.4 Digital Thermal Sensors (DTS)

The on-die Digital Thermal Sensor (DTS) reports the temperature of the associated core as a temperature relative to the factory configured TCC activation temperature (i.e., REF_TEMP or T_{J-MAX}). There is one DTS per core.

The DTS outputs a temperature (IA32_THERM_STATUS[dig_temp_readout]) relative to the maximum supported operating temperature of the SoC which is the PROCHOT_B signal activation temperature as configured when the SoC was manufactured. The factory configured TCC activation temperature can be read from the CPU_THERM_TEMPERATURE[ref_temp].

The system BIOS can convert the relative temperature to an absolute temperature. The temperature returned by the digital sensor is dependent on the configuration of CPU_THERM_CFG2[therm_valid_range].

The SoC is out of specification anytime the temperature is above PROCHOT. Extreme over temperature conditions are detectable via an Out Of Spec status bit. When this bit is set, the processor is operating out of specification and an immediate shutdown of the system occurs. The system BIOS detects that this bit is set and informs the operating system that a critical shutdown is warranted. The CPU operation and code execution are not guaranteed once the activation of the Out of Spec status bit is set.

For more information on the conversion of the DTS relative temperature to an absolute temperature, refer to the *Intel® Atom™ Processor C2000 Product Family BIOS Writer's Guide (BWG)*.

8.5 Thermal Interrupts and Thresholds

When one of the specified thermal thresholds is exceeded, the threshold is programmed to trigger an interrupt. The thermal management task in P-Unit code monitors the die temperature readings from each sensor, detect if any interrupt thresholds are exceeded and enabled, and write the bits in the via PMLink configuration registers. The thermal thresholds in the SoC are both programmable via software and hard coded. Table 8-3 summarizes the applicability of the thermal thresholds and their supported actions and triggers.

The SoC supports additional thermal-related mechanisms that are not directly tied to thermal sensors.

Table 8-3. Thermal Threshold Descriptions and Actions

Sensor Location	Thermal Trigger	Control/Description	Actions
Core	Programmable Threshold 1 (per core)	Threshold specified as an offset below T_{J-MAX}	ACPI/software use
Core	Programmable Threshold 2 (per core)	Threshold specified as an offset below T_{J-MAX}	ACPI/software use
Core	HOT Threshold	Threshold specified as an offset below T_{J-MAX}	• Intel® Thermal Monitor 1/ Intel® Thermal Monitor 2 throttling • PROCHOT_B pin assertion if enabled
Core	Out of Specification Threshold	Threshold specified as an offset above T_{J-MAX}	ACPI/software use
Uncore	Programmable Threshold 1 (AUX0)	Threshold specified as an offset below T_{J-MAX}	ACPI/software use
Uncore	Programmable Threshold 2 (AUX1)	Threshold specified as an offset below T_{J-MAX}	ACPI/software use
Uncore	Programmable Threshold 3 (AUX2)	Threshold specified as an offset below T_{J-MAX}	ACPI/software use
Uncore	Programmable Threshold 4 (AUX3/ HOT Trip)	Threshold specified as an offset below T_{J-MAX}	ACPI/software use
System (Core and Uncore)	PROCHOT_B	System triggers Intel® Thermal Monitor via PROCHOT_B.	• Intel® Thermal Monitor 1/ Intel® Thermal Monitor 2 throttling • PROCHOT_B interrupt
System (Core and Uncore)	THERMTRIP_N	Threshold specified as an offset above T_{J-MAX}	Shut off all the PLLs and power rails to prevent damage.

8.5.1 Core Programmable Thresholds

Two programmable thresholds per CPU core sensor are provided for Advanced Configuration and Power Interface (ACPI) compliant thermal management via software. The thresholds are specified as an offset below $T_{J\text{-}MAX}$ via registers and configured to trigger a CPU thermal interrupt. ACPI software routines then take appropriate action when the interrupts get serviced.

8.5.2 Core HOT Threshold

The Core HOT threshold indicates that a thermal sensor in the core has exceeded the maximum-allowed operating temperature (defaults to $T_{J\text{-}MAX}$). The die temperature readout of each sensor is periodically compared against the $T_{J\text{-}MAX}$ value. Corrective actions are taken if the threshold value is exceeded. The Core HOT condition triggers the Intel® Thermal Monitor 1 or Intel® Thermal Monitor 2 thermal control circuit mechanisms (if enabled).

Programmable hysteresis offsets are utilized to prevent oscillation of the Core HOT condition around the threshold. Hysteresis in terms of temperature offset is specified for both triggering and deactivating the Core HOT condition.

8.5.3 Core Out of Specification Threshold

The Core Out of Specification threshold indicates that a thermal sensor is close to reaching a catastrophic temperature which permanently damages the device. This programmable threshold is set above $T_{J\text{-}MAX}$ with a default value of 5 °C above $T_{J\text{-}MAX}$. This threshold does not trigger any thermal throttling mechanisms.

8.5.4 Uncore Programmable Thresholds

For the uncore, the programmable trips are programmed to cause different actions when triggered to reduce die temperature.

8.5.4.1 Aux3 Trip

By default Aux 3 (Hot Trip) point is set by fuses, but the software/firmware has an option to set these to a different value.

This trip point is enabled by P-Unit during the M1 stage. The P-unit monitors and controls the system temperature while the rest of the system is being setup.

8.5.4.2 Aux2, Aux1, Aux0Trip

These are fully programmable trip points for general hardware protection mechanisms. The programmable trips are only active after the software/firmware enables the trip.

Note: Unlike Aux 3, the rest of Aux trip registers are default to zero. To prevent spurious results, the software/firmware programs the trip values before enabling the trip point.

8.5.5 PROCHOT_B

The Core HOT trip and the uncore AUX 3 trip signals are individually sent to P-Unit code, which internally combines them and drives the appropriate PROCHOT_B signal. Bi-directional PROCHOT_B allows system assertion of the PROCHOT_B input to trigger Intel® Thermal Monitor 1 or Intel® Thermal Monitor 2 throttling mechanisms if they are enabled. One example of bi-directional PROCHOT_B usage is activation of TCC when a voltage regulator high current condition is detected.

The platform board must ignore this SoC output signal while PMU_PLTRST_B (active-low SoC output) is asserted.

8.5.6 MEMHOT_B

The platform board must ignore this SoC output signal (MEMHOT_B) while PMU_PLTRST_B (active-low SoC output) is asserted.

8.5.7 THERMTRIP_N Signal

In the event of a catastrophic cooling failure, the processor automatically shuts down when the silicon temperature reaches its operating limit. At this point the system bus signal THERMTRIP_N becomes active and power must be removed from the processor. The THERMTRIP_N activation is independent of the processor activity and does not generate any bus cycles. Refer to the *Intel® Atom™ Processor C2000 Product Family Thermal and Mechanical Specifications and Design Guidelines* for additional details.

The temperature when the THERMTRIP_N signal becomes active is individually calibrated during manufacturing. The temperature at which THERMTRIP_N becomes active is roughly parallel to the thermal profile and greater than the PROCHOT_B activation temperature. Once configured, the temperature at which the THERMTRIP_N signal is asserted is neither re-configurable nor accessible to the system.

The platform board must ignore this SoC output signal while PMU_PLTRST_B (active-low SoC output) is asserted. At the time that THERMTRIP_N is asserted, the system automatically begins shutdown of the system. The period of THERMTRIP_N assertion is around 200 ns, but the duration is not guaranteed.

8.6 Processor Thermal Control Circuit (TCC) Mechanisms

Thermal Control Circuit (TCC) mechanisms are implemented to reduce temperature by reducing power consumption in response to a Core HOT condition. The core implements Intel® Thermal Monitor 1 (TM1) clock modulation similar to legacy devices and Intel® Thermal Monitor 2 (TM2) core frequency/voltage reduction. All thermal control circuit mechanisms are controlled via P-Unit routines.

8.6.1 Clock Modulation (Intel® Thermal Monitor 1)

Intel® Thermal Monitor 1 (TM1) effectively stops the core clock periodically to reduce processor power consumption. The duration for which the clock is modulated is programmable within each core register. The on/off duty-cycle is adjusted in increments of 12.5% and must be configured by the BIOS.

Note: The stop-clock duration is frequency dependent and higher frequencies stop the clock for shorter durations for a given duty-cycle configuration.

TM1 is enabled via each core register and activated as a secondary control mechanism if TM2 (described below) is unsuccessful at reducing processor temperatures for a period of time.

8.6.2 Core Frequency/Voltage Reduction (Intel® Thermal Monitor 2)

The core implements an adaptive Intel® Thermal Monitor 2 (TM2) mechanism which transitions to a lower operating frequency and voltage Low Frequency Mode (LFM). The core implements this mechanism only if higher performance frequency and voltage points are not successful in reducing temperatures (i.e., chooses the highest performance operating point which reduces processor temperatures).

Intel® Thermal Monitor 2 is automatically selected as the primary thermal control mechanism and selection of LFM versus adaptive mode is achieved via the thermal configuration registers.

8.6.3 Thermal Status

Thermal trip events are captured in status registers. Associated with each event is a set of programmable actions. For a complete list of refer to the *Intel® Atom™ Processor C2000 Product Family BIOS Writer's Guide (BWG)*.

8.7 Memory Thermal Control

Two mechanisms for managing memory temperatures are available: the memory bandwidth counter and the memory module temperature monitoring. Since the SoC has no mechanism to determine the temperature of the memory modules, the BMC, or other platform-board external circuitry, is needed to provide this information via PECI over the SMBus.

8.7.1 Memory Bandwidth Counter

Memory event based bandwidth throttling is available as a fallback thermal protection feature if external thermal sensors are not available. Memory reads and writes are counted and compared to a threshold trip point to reduce memory bandwidth when needed to reduce memory module temperatures. These counter-based trip points are enabled using the thermal management control registers.

8.7.2 Memory Temperature Monitoring

Open Loop Thermal Throttling (OLTT) and Closed Loop Thermal Throttling (CLTT) pass-through is supported on the SoC to help optimize platform power/acoustics. The SoC supports DDR3 DRAM technology. The temperature sensor on DIMM (TSOD) is required for CLTT. OLTT is also supported. The implementations of memory thermal throttling are defined as follows:

1. Open Loop Thermal Throttling: The system does not change any of the control registers in the memory controller during runtime. OLTT control registers are configured by BIOS MRC and remain fixed after post.
2. Closed Loop Thermal Throttling (pass-through): The system does not change any of the control registers in the memory controller during runtime. CLTT pass-through control registers are configured by BIOS MRC and firmware and remain fixed after post. The memory controller does not poll the DIMMs directly, so this mode still requires PECI over SMBus pass-through mode as described below.
   a. PECI over SMBus pass-through: There is a PCU pass-through feature in which the BMC or other external controller can write temperature data to the PCU using PECI for platforms where the TSOD is not available.
   b. The following temperature ranges are defined for the CLTT pass-through Memory throttling:
   — 82 °C - Unconstrained performance, set throttling to peak.
   — 93 °C - Set throttling to 10% of peak.
   — 100 °C - Set throttling to 1% of peak to attempt to prevent system shutdown.

§ §

9 Power Management

The SoC is a server-class component and provides dynamic power management that fit a number of usages like storage and networking. As the technology changes and the number of cores per node and per module changes, this places a requirement to find the best approach to manage the TDP power of the node/module with respect to the available power budget.

9.1 Overview

The power management control is comprised of a number of Intel® proprietary mechanisms. The power management signal interface to the rest of the platform is through the Power Management Controller (PMC) block in the Platform Controller Unit (PCU). In some areas, this chapter refers to this distributed function as the SoC Power Management Unit (PMU).

An external, board-level Baseboard Management Controller (BMC) or some other embedded controller, is required to manage the platform power planes, power-on, sleep states, and reset signaling. This chapter refers to the BMC or Embedded Controller (EC). The EC interfaces with the SoC internal fabric to perform its management functions.

The SoC power management is responsible for the following tasks:

- Managing the internal power wells voltages
- Communicating with the EC
- Resetting sequencing
- Managing processor C-states
- Managing L2 cache dynamic sizing
- Managing Sleep-state entry sequences
- Managing DDR3 power management and RComp routines
- Controlling Low Pin Count (LPC) interface clock
- Directing processor thermal management
- Interfacing with the BIOS and the operating system software
- Managing Intel® Turbo Boost Technology and RAPL control

Table 9-1. References

Reference	Revision	Date	Description
ACPI Specification	5.0	DEC 2011	*Advanced Configuration and Power Interface Specification*, Revision 5.0
VR12/IMVP7 Specification	1.61	OCT 2011	*VR12/IMVP7 Pulse Width Modulation*, Revision 1.61 (Intel Document Number 397113)
VR12/IMVP7 Protocol	1.5	AUG 2010	*VR12/IMVP7 SVID Protocol*, Revision 1.5 (Intel Document Number 456098)

9.2 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The signal/pin name
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

See Chapter 31, "Signal Names and Descriptions" for details of these signals.

Table 9-2. sVID Controller Signals

Signal Name	Direction/ Type	Description
SVID_DATA	I/O OD	**sVID Data:** Used by the SoC to send requests and data to the external Voltage Regulator (VR) and then by the VR to respond.
SVID_CLK	O OD	**sVID Clock:** sVID requests driven on SVID_DATA by the SoC use this clock. The VR uses this clock to register-capture the requests. When the VR responds with data on SVID_DATA, the VR also uses this SoC-driven clock.
SVID_ALERT_B	I OD	**sVID Alert:** Used by the VR to signal to the SoC that the prior request has not reach the requested operating point.

9.3 Power Management Features

The power management features are:

- ACPI system power states supported: G0 (S0), G2 (S5), G3 (Mechanical Off)
  - G1 (S1, S2, S3, S4) are not supported.
  - The SoC has some G1 capabilities, but must not be used. They are not supported by Intel.
- ACPI processor (CPU) C-states: C0, C1, C6C. The C4 state is not supported.
- ACPI device states: D0, D3
- PCI Express*: L0 and L1 Supported (L0s not supported).
- Enhanced Intel SpeedStep® Technology functionality on CPU local bus
- Enhanced Intel SpeedStep® Technology
- Hardware throttling
- Clock gating
- Thermal throttling
- Dynamic I/O power reductions (disabling sense amps on input buffers, tri-stating output buffers)
- Re-programmable Power Management Unit (PMU)
- PECI over SMBus
- Running Average Power Limiter (RAPL)
- DDR3 SDRAM memory controller and PHY:
  - Dynamic rank power down.
  - Dynamic power down is employed during normal operation. If all the pages have all been closed at the time of CKE pin deassertion, the SDRAM devices enter the pre-charge power-down state. Otherwise the devices enter the active power-down state.
  - Conditional memory self-refresh.
  - DLL master/slave shut down based on CPU state.
  - Address and Command signal tri-state when all memory is in power down or self-refresh, or when not in use (no chip select asserted).
  - Chip-select tri-state for a powered-down row.
  - Clock tri-stating for unpopulated DIMMs.
  - CKE/CS tri-stating for unpopulated rows.
  - Conditional memory self-refresh during C6.
  - Conditional and software-directed memory self-refresh.
    Supports conditional self-refresh entry in the C6 state based on memory request traffic from the host interface agents.
- Debug and testability hooks

9.4 Internal Power Wells

The SoC internal circuitry is powered by three power wells:

- Core power well
- SUS power well
- RTC power well

9.4.1 Core Power Well

This power group includes all internal voltage rails and associated power wells that are on only when the system is in the system sleep S0 state (system is fully powered-on). These voltage rails are turned off when the system transitions to one of the other low-power system sleep states. See Figure 9-1 for the ACPI states flow diagram.

9.4.2 SUS Power Well

This power group includes internal voltage rails and associated power wells that are on when the system is in the S0 and S5 states. These voltage rails are turned off when the system transitions from S5 to the G3 (mechanically off) state.

Most of the power management signal pins have their drivers/receivers in the SUS power well.

9.4.3 RTC Power Well

This power group includes all internal voltage rails and associated power wells that are always on, even when the system is in the G3 (mechanical off) state. This group is supplied its 3.3V from the SUS power supply. When the system is in the G3 state, the RTC power well is powered from an external battery source, typically a 3.0V lithium-type coin cell.

If a complete power failure (no AC power and no battery back-up supply) occurs, this voltage rail does not provide any power if there is no functioning battery providing its power.

Note: In the G3 state (mechanical off), it is permissible for designs to not have an external coin-cell battery if the design does not need to preserve information when the system power is turned off.

9.5 Supply Voltage Rails

The SoC is a highly integrated component, where many traditional subsystems are contained on one die. No on-die voltage regulator exists except for power supplied to the internal clock generators and thermal sensors.

A total of 13 external unique rails need to be supplied. Table 9-3 shows the voltage rail of each external Voltage Regulator (VR), its voltage levels and a list of internal units associated with the voltage rail.

Table 9-3. SoC Voltage Rails

Typical Platform Voltage Source	VR Rail	Voltage	SoC Internal Units	S0	S5	RTC
Coin Cell Battery	VRTC3P0	3.0V	Real Time Clock (RTC)	On	On	On
sVID VCC VR	VCC	SVID 0.5-1.3V	CPU Core	On	-	-
sVID VNN VR	VNN	SVID 0.5-1.3V	VNN	On	-	-
V1P0	V1P0A Always On	1.00V	• GPIOSUS signal pins • GbE control and I/O • SUS circuitry in PCU • Power Management Controller (PMC)	On	On	-
V1P0	V1P0S Switched	1.00V	• PCI Express* Root Ports • SATA Controllers • USB 2.0 I/O • Internal Clock Gen • GPIOCORE signal pins • DDR3 I/O	On	-	-
V1P07 Switched	VCCSRAM	1.07V	• Core L2 Cache	On	-	-
V1P8	V1P8A Always On	1.8V	• GPIO SUS • USB 2.0 SUS	On	On	-
V1P8	V1P8S Switched	1.8V	• GPIO HV Core intermediate	On	-	-
sVID_VDDQ_A	VDDQA _1P5 _1P35	1.5V 1.35V	DDR I/O (Channel 0)	On	-	-
sVID_VDDQ_B	VDDQB _1P5 _1P35	1.5V 1.35V	DDR I/O (Channel 1)	On	-	-
V1P35	V1P35S Switched	1.35V	Internal debug and test	On	-	-
V3P3	V3P3A Always On	3.3V	• GPIO SUS-Well High-Voltage signal pins • USB 2.0 SUS signal pins	On	On	-
V3P3A	V3P3S Switched	3.3V	• GPIO Core-Well High-Voltage signal pins	On	-	-

The VCC, VNN, VDDQA, and VDDQB rails are sVID-based variable rails and change dynamically. Eight of the 13 VR Rails in Table 9-3 follow the naming convention of including a suffix indicating the power modes in which the rail is active.

- A - Always On (AON). Remains powered during S0-S5 states.
- S - Switched. Remains powered during S0 only.

9.6 Serial Voltage Identification (sVID) Controller

The sVID controller consists of three signal pins and is defined in the *VR12/IMVP7 Pulse Width Modulation*, Revision 1.61. The three signals provided by the SoC are in Table 9-2.

9.6.1 SVID VR Requirements

It is required that the voltage rails for the cores (VCC) and fabric (VNN) support IMON as defined by the VR12/IMVP7 SVID protocol specification.

9.6.1.1 SVID Controller Addressing Requirements

The following address assignments are required. This definition supersedes any definitions defined in the *VR12/IMVP7 Pulse Width Modulation*, Revision 1.61 specification.

SoC Power Management expects VR addresses to always be sequential with no gaps in address assignments for any Voltage Rails.

Table 9-4. SVID Controller Addressing Requirements

Rail	Address[3:0]	Definition
VCC	0h	CPU Core rail
VNN	1h	SoC rail
VDDR_A	2h	DDR Channel A (only if VID is used on platform). Soft strap must be set to indicate whether rail is present. See Section 16.4, "Soft Straps" on page 362.
VDDR_B	3h	DDR Channel B (only if VID is used on platform). Soft strap must be set to indicate whether rail is present. See Section 16.4, "Soft Straps" on page 362.

9.6.2 Command Byte Encoding

Refer to *VR12/IMVP7 SVID Protocol*, Revision 1.5 for protocol details.

9.6.2.1 sVID Commands

The sVID commands are shown in Table 9-5. The send payload VID values are shown in the Table 9-13.

Table 9-5. sVID Commands

Command	Command Code	Send Payload	Receive Payload	Description
Not Supported Reserved	00h	(Extended Command)		
SetVID_Fast	01h	VID	n/a	Sets the new VID target. VR Jumps to the new VID target with controlled (up or down) slew rate programmed by the VR. When the VR receives a VID Moving Up command it exits all low-power states to the normal state to ensure the fastest slew to the new voltage. VR sets VR_settled bit and issues alert when VR has reached new VID target.
SetVID_Slow	02h	VID	n/a	Sets the VID target. VR Jumps to new VID target with controlled slew rate (up or down) programmed by the VR. SetVID-Slow is 4x slower than SetVID-fast. When VR receives a VID Moving Up command it exits all low-power states to the normal state to ensure the fastest slew to the new voltage. VR sets VR_settled and issues alert when VR has reached new VID target.
SetVID_Decay	03h	VID	n/a	Sets the VID target. VR jumps to new VID target, but does not control the slew rate, the output voltage decays at a rate proportional to the load current. SetVID_Decay is only used in VID down direction. VR sets VR_settled bit is set, but alert line is not asserted for SetVID-decay.
SetPS	04h	Power State (PS)	n/a	Sends information based on the CPU power state to the VR controller so it configures the VR to improve efficiency, especially at light load.
SetRegADR	05h	VR Register Address	n/a	Sets the address pointer in the data register table. Typically the Next command SetRegDAT is the payload that gets loaded into this address. However, for multiple writes to the same address, only one SetRegADR is needed.
SetRegDAT	06h	VR Register Data	n/a	Writes the contents to the data register that was previously identified by the address pointer with SetRegADR.
GetReg	07h	VR Register Address	VR Register Data	Slave returns the specified register contents as the payload. The majority of the VR monitoring data is accessed through the GetReg command.
Test_Mode	08h			Vendor Defined
Reserved	08h through 1Fh			Slave returns a Reject acknowledgment.

9.7 Active State Power Management Overview

When one or more of the processor cores are active, the SoC power management adjusts the operating conditions as needed to reflect both the objectives of the operating system and the physical temperature and power-related constraints. In most cases, objectives are met by adjusting the target frequency of each active core. The SoC power management adjusts each core target clock speed to a discrete operating point. The discrete steps between these operating points correspond to core clock ratios.

As the target frequency is changed, the operating voltage is set to the highest sVID request from all modules. Several factors contribute to the active operating point of a core at any given instant:

- Operating System (OS) P-State requests for that core
- P-State requests of the other cores in the package
- The P-State selection model
- The current temperature of each core
- Any available turbo modes
- Previous P-states and C-states
- Any other factors

Based on a snapshot of these factors, the SoC determines a target ratio and associated frequency for each core as well as a package target. The resolved package operating point is the highest requested operating point of any of the cores in the package, after all factors are taken into account.

When the core is idle (in a lower-power C-State than C0), its voting rights may be suspended based on the Energy Performance BIAS MSR setting (0x1b0), and the core either acts as a slave to the operating voltage specified by other cores or has its voltage removed by turning off its power gate.

When resolving the operating point of an individual core, the various contributing factors are weighed independently to come up with three ratio targets:

- Software-initiated request (OS P-State request),
- Thermally-constrained operating target, and
- Power-constrained operating target.

The upper and lower bounds of these targets are weighed and prioritized to come up with a final ratio resolution for each core and the sections.

Additional active-state power management features change the power processor core characteristics without changing the operating voltage and frequency. For the most part, these features provide legacy power management functionality, and are superseded by the newer voltage and frequency scaling capabilities of today’s processors.

9.8 System Global Power States

Figure 9-1 shows the system Global Power States as defined by the *Advanced Configuration and Power Interface Specification*, Revision 5.0. Table 9-6 shows the power states defined for platforms based on the SoC.

Figure 9-1. Global System Power States and Transitions

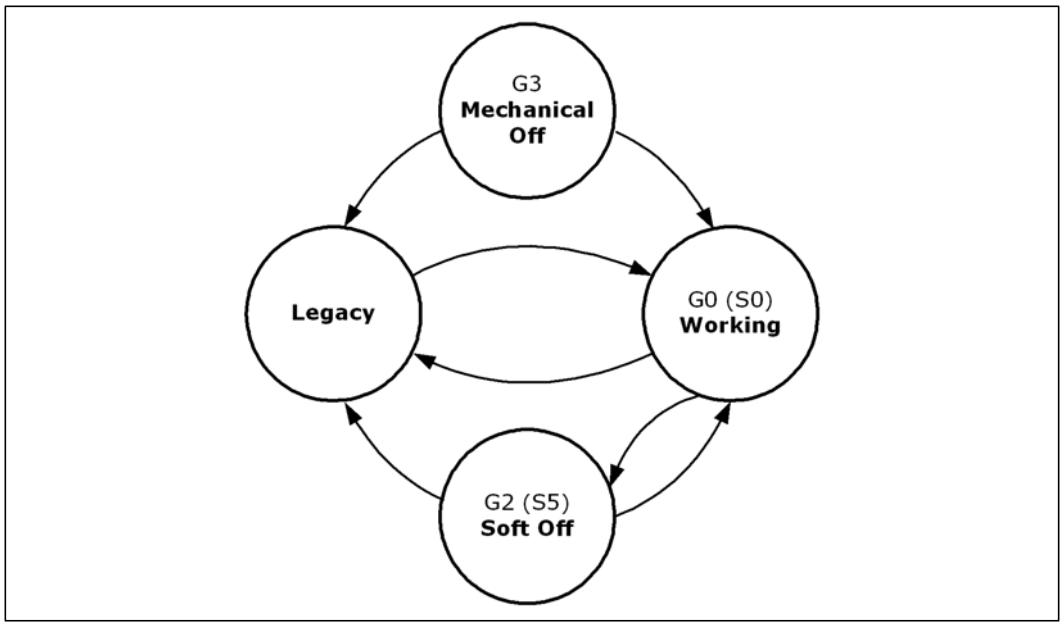


Table 9-6. ACPI Power States

State (Global/Sleep/CPU)	APCI Name	Description
G0/S0/C0	Working (Full On)	CPU operating. Individual devices are shut-off to save power.
G0/S0/C1 G0/S0/C6	Working with CPU Power Management	The different CPU operating levels are defined by Cx states.
G2/S5	Soft Off	System context not maintained. All power is shut except for the logic required to restart. A full boot is required when waking.
G3	Mechanical Off	• System context not maintained. • All power shut except for the RTC. • No Wake events occur, because the system does not have any power. This state occurs if user removes the batteries, turns off a mechanical switch, or if the system power supply is at a level that is insufficient to power the waking logic. When system power returns, transition depends on the state before entry to G3.

Table 9-7. ACPI Power State Transitions for the SoC

State (Global/Sleep/CPU)	Transition Trigger	Next State
G0/S0/C0	Executing the MWAIT instruction or LVL Rd	G0/S0/Cx
	Sleep Enable (SLP_EN) bit written to 1 by the software in the Power Management 1 Control (PM1_CNT) register	Specified by the 3-bit code Sleep Type (SLP_TYP) field of the PM1_CNT register: 000 G0/S0 - Working (Full On) 001 G1/S1 - Not supported by the SoC 010 Reserved 011 Reserved 100 Reserved 101 G1/S3 - Suspend-To-RAM - Not supported by the SoC 110 G1/S4 - Suspend-To-Disk S4 - Not supported by the SoC 111 G2/S5 - Soft Off
	Power Button Override	G2/S5
	Mechanical Off/Power Failure	G3
G0/S0/C1 G0/S0/C6	C-State break events including: • CPU Snoop • MSI • Legacy Interrupt • Always-On (AONT) Timer expires	G0/S0/C0
	Power Button Override	G2/S5
	SUS Well Power Failure	G3
G2/S5	Any Enabled Wake Event	G0/S0/C0
	SUS Well Power Failure	G3
G3	Power Returns	Option to go to: • S0/C0 (Reboot) • G2/S5 (Stays off until the power button is pressed or other enabled wake event)
S0/G3/S0	Surprise Power Loss	Not supported. No support within the SoC to maintain data integrity during unplanned power loss.

9.8.1 Low-Power S0 Idle

The SoC implements proprietary hardware to minimize power consumption when the platform is in use, but idle. The platform achieves power savings while in the G0/S0 (full on) state similar to or better than typically achieved in the G1/Sx (sleeping) power states.

ACPI provides a mechanism for the platform BIOS to indicate to the Operating System Power Management (OSPM) that such capability is available. Refer to the *Advanced Configuration and Power Interface Specification*, Revision 5.0 for details.

When this capability is available, the OSPM keeps the system in S0 idle for its low-latency response and its connectedness rather than transitioning to a system sleep state which has neither.

While the SoC implements S0 idle mechanisms, the SoC does not support OS-directed, centrally-controlled, power savings commonly referred to as S0ix mechanisms.

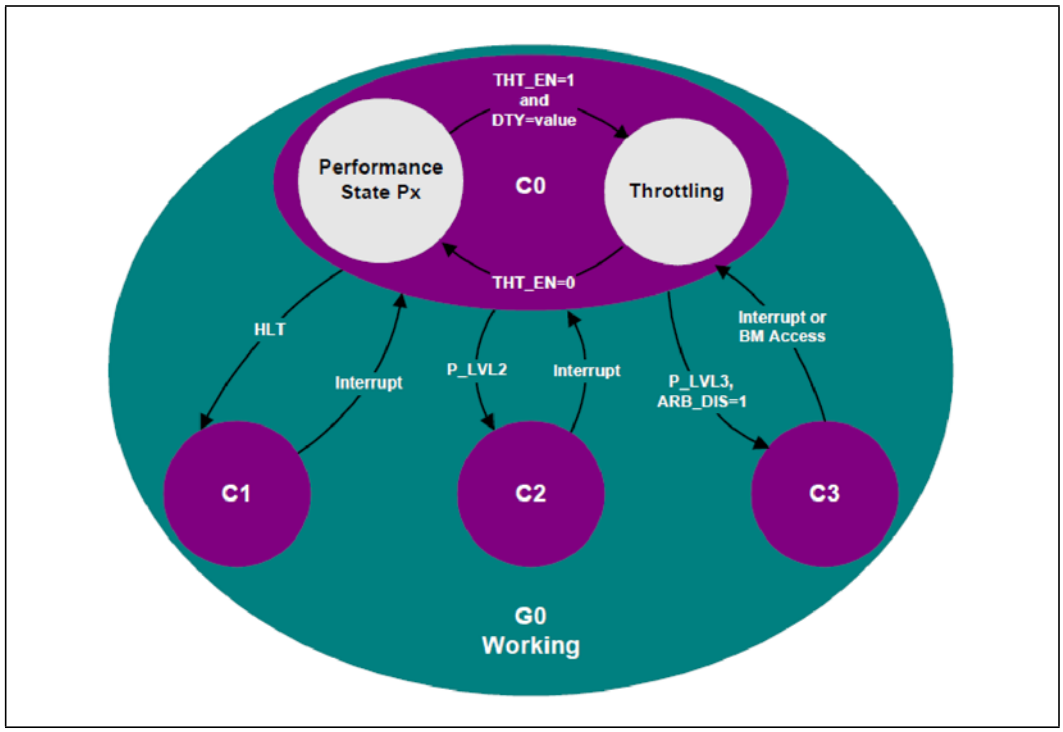
Some form of S0 idle occurs when the system is quiet, for example when the system is waiting for user input or when drivers are not causing unnecessary CPU or SoC activity. To conserve power, the SoC does gate clocks and powers down to a number of the internal blocks and external-pin I/O buffers.

9.9 Processor Power States - C-States

Figure 9-2 shows the Processor Power States described in the *Advanced Configuration and Power Interface Specification*, Revision 5.0.

The processor Performance States (P-States) are described later in "Processor Performance States - P-States" on page 165.

Figure 9-2. Processor Power States



Concerning the SoC CMP Module and its interface to the System Agent, the SoC has three types of C-States:

- Core C-States. See Table 9-8.
- Module C-States. See Table 9-9.
- Package C-States. See Table 9-10.

Module C-States (designated as MC) pertain to the two-core CMP module while the Core C-States (designated as CC) independently pertain to each of the two cores within the CMP. Package C-States (designated as PC) pertain to the entire set of CMP modules available in the particular product SKU.

Table 9-8. Core C-States

General C-State	Core C-State	Description
C0	C0	Active State
C1	CC1	Some core clocks gated. L1 Data Cache Snoops are serviced.
C6	CC6-NS	• Cores are placed in LFM (based on Energy Perf Bias Setting) • Core power gated and clock gated • L1 Data cache is flushed *The NS hints from the software.*

Table 9-9. Module C-States

General C-State	CMP Module C-State	Core Status	L1 and L2 Cache Status
C0	MC0	At least one core in C0 state	Normal L1 and L2 cache operation
C1	MC1	Both cores HALTed *Most clocks off*	No cache flushed Cache Snoops wake-up cores
C6	MC0	Both cores are in the C6 (powered-off) state. VID is determined at a package level not module level. *CPU reference clock off*	Core L1 data cache flushed Four of 16 ways of L2 cache retained

Table 9-10. Package C-States

General C-State	CMP Package C-State	When Entered
C0	PC0	When in PC4, PC6, or PC7, one of the following occurs: • MSI Break • Snoop Wake • Machine Check Error • Always-On Timer (AONT) expires in one of the cores Various actions are taken by the SoC once the PC0 state is entered.
C1	PC1	When all cores are in C1 and based on the overall C6 residency, the frequency will be lowered to a value between LFM and Guaranteed.

The C-State characteristics for the Silvermont processor are different than previous Intel® Atom™ processors. Most changes involve the C6 and the new C6C state. The SoC core C6 state does not provide flushing of dirty data from the L2 cache.

Note: C1E cannot be disabled because it is required for reliability purposes. This means that when all the cores are idle for a period of time, the SoC will lower the frequency to the low frequency mode, particularly if C6 is disabled.

9.10 Performance States

This section describes the concept of processor and device performance states. Processor and device performance states (Px states) are power consumption and capability states within the active/executing states, C0 for processors and D0 for devices. Performance states allow the Operating System Power Management (OSPM) to make trade-offs between performance and energy conservation. Processor and device performance states have the greatest impact when the states invoke different device and processor efficiency levels as opposed to a linear scaling of performance and energy consumption. Since performance state transitions occur in the active/executing device states, ensure that performance state transitions do not adversely impact the system.

Disabling the software from requesting P-States is possible by setting bit 16 in IA32_MISC_ENABLE. This does not prevent the SoC changing frequency in voltage for thermals, RAPL, and PkgC1E.

9.10.1 Processor Performance States - P-States

The SoC supports P-States for every dual-core pair within a module. Based on power performance analysis, the SoC only support Package Level P-States. The Power Management Unit will select the highest P-state from all requests across all modules and apply that state to all cores. The internal power management sets a lock bit to ensure that the cores are always at the same P-State.

9.10.1.1 Frequency/Voltage Scaling

SoC CPU supports P-States for OS-controlled management of processor performance. The CPU range of operation is broken down into P-States and T-States.

Operating systems which support ACPI may utilize the BIOS FADT table to map ACPI P-States. Table 9-11 describes ACPI P-State mappings.

Table 9-11. ACPI P-State Mappings

P-State	ACPI Meaning	Frequency Mapping
P0	Performance is preferred over power efficiency	Greater-than or equal-to the Maximum Non-Turbo Limit Ratio. This is set based on thermal/electrical limits, platform constraints, and other parameters.
P1	Maximum performance/efficiency is desired	Maximum Non-Turbo Limit Ratio (Guaranteed Ratio)
P2 through PN	Intermediate performance/efficiency is desired	Less-than the Maximum Non-Turbo Limit Ratio, Greater-than the Maximum Efficiency Ratio
PN	Maximum performance efficiency is desired (best for average power)	Maximum Efficiency Ratio correlates to minimum operational voltage (LFM_RATIO)

The OS requests a P-State based on application performance needs. A desired P-State is requested via IA32_PERF_CTL (CLOCK_CR_GEYSIII_CONTROL). The SoC supports the Enhanced Intel SpeedStep® Technology.

9.10.2 Software P-State Requests

A P-State is a software-visible frequency/voltage operating point. The OS, BIOS, or any Ring-0 software has the permissions to make P-State change requests.

9.10.2.1 Windows 7: P-State Transitions with ACNT/MCNT

- P-States are evaluated every 100 ms (configurable)
- Transitions are every 100-300 ms (configurable)
- Uses processor utilization and ActualCount/MaxCount (ACNT/MCNT) feedback if available
- P-State TARGET = %Busy * ACNT/MCNT
- ACNT and MCNT counters reset each sampling period (same as idle accounting)
- Uses pre-calculated increase/decrease levels

9.11 Power Management Technologies

9.11.1 Intel® Turbo Boost Technology

Intel® Turbo Boost Technology enables higher performance through the availability of increased core frequency under certain configurations and workloads. Turbo allows processor cores to run faster than the specified operating frequency if the processor is operating below rated power, temperature, and current specification limits of the system. Turbo is engaged with any number of cores or logical processors enabled and active, enabling increased performance of both multi- and single-threaded workloads.

The BIOS enables/disables turbo features. The power-on default value of IA32_MISC_ENABLES[38] indicates to the BIOS if the turbo features are present. A default value of 1 indicates that turbo features are present and disabled. The BIOS clears the IA32_MISC_ENABLES[38] to 0 to enable turbo. A reset-default value of 0 indicates that turbo features are disabled and not available.

The Operating System (OS) and applications must use CPUID.06H:EAX[1] to detect whether the BIOS has enabled turbo features. If IA32_MISC_ENABLES[38] is set, CPUID.06H:EAX[1] returns 0.

Because IA32_MISC_ENABLES[38] is defined per-package, CPUID has to read from the uncore to get MISC_ENABLES[38]. The OS or BIOS manages read-modify-write conflicts cross-core. Setting IA32_MISC_ENABLES[38] on any core causes the SoC to disable turbo operation for ALL cores.

Certain software workloads may not be able to tolerate the non-deterministic aspects of turbo operation. The software temporarily disables turbo operation by setting CLOCK_CR_GEYSIII_CONTROL[32] bit of the IA32_PERF_CTL Model-Specific Register (MSR) [MSR 0199h]. As previously mentioned, disabling turbo on any core causes turbo to be disabled for ALL cores.

Configuration of certain turbo-power budget settings is accessible by the OS/driver software via SoC Sideband (SB) registers. PKG_TURBO_POWER_LIMIT configuration registers are duplicated in the SB register space for this purpose. The MSR copies of the registers are initialized by the BIOS to typical recommended settings and are overridden with more conservative values by the OS/driver by programming the corresponding SB registers. The SoC power management reads both copies of the registers and applying the more restrictive settings to the turbo algorithms.

9.11.1.1 Voltage Regulator Constraints

For some platform, the voltage regulator for the shared VCC voltage rail has some maximum current limits. A typical voltage regulator for a platform using the SoC has a Thermal Design Current (I_{TDC}) specification and a Maximum Current (ICC_{MAX}) limit.

I_{TDC} represents the current which is sustained indefinitely by the voltage regulator (i.e., able to sustain a TDP workload on all processors running at warranted frequency), whereas the ICC_{MAX} is the peak current which the voltage regulator does source without tripping any protective circuitry. Any current greater than I_{TDC} is only sustained for a short duration, with ICC_{MAX} only sustained typically for around 10 ms.

Such voltage-regulator specifications must be taken into account when configuring the aggressiveness of turbo operation for the device. Turbo operating points are limited based on the number of active cores to keep the current drawn within the ICC_{MAX} limit.

Refer to the *Intel® Atom™ Processor C2000 Product Family - Platform Design Guide (PDG)* for additional details and design guidance.

9.11.1.2 Thermal Design Power Constraints

The Thermal Design Power (TDP) represents the power consumed by the device when running a realistic worst-case workload at T_{J-MAX} temperature. The realistic-worst-case workload is determined based on knowledge of the target application and realistic usage scenarios. The system thermal solution must be designed to dissipate the heat generated during a sustained period of activity with all cores running at TDP at guaranteed maximum frequency.

Refer to the *Intel® Atom™ Processor C2000 Product Family Thermal and Mechanical Specifications and Design Guidelines* for additional details and design guidance.

9.11.2 Running Average Power Limiting (RAPL)

The SoC contains proprietary power monitors and Running Average Power Limit (RAPL) algorithms that calculate an energy budget and convert the budget into voltage/ frequency working points.

The SoC supports RAPL control through these SoC interfaces:

- Memory-Mapped I/O (MMIO) interface for drivers to program RAPL limits and monitor RAPL performance.
- Platform Environment Control Interface (PECI) via SMBus for the platform firmware to program RAPL limits and monitor RAPL performance.
- I/O Port CF8/CFC for registers not mapped to MSR or MMIO space.

9.11.3 Always-On Timers (AONT)

Always-On Timers run while the CPU/SoC is in the S0 state, including the S0 idle state, and are used to periodically wake-up the cores from sleeping. They do not run when the CPU transitions out of S0.

9.11.4 I/O Device Controller Enable/Disable

Table 9-12. I/O Power Management Summary

Integrated I/O Device	I/O Feature Not Used by Customer	Nothing Connected to Interface at Boot Time	PC6 Idle (S0idle)
PCIe* Root Ports	The BIOS disables the circuitry and power on a per-lane basis.	The BIOS disables the circuitry and power on a per-lane basis.	Depending on the product SKU, internal clocks off and power gated from controller.
SATA2	The BIOS disables the circuitry and power-on a per-port basis.	Disabled at power-on unless explicitly enabled by the software.	Depending on product SKU, disabled and power gated from controller or internal clocks off.
SATA3	The BIOS disables the circuitry and power-on a per-port basis.	Disabled at power-on unless explicitly enabled by the software.	Depending on product SKU, disabled or internal clocks off.
GbE	The BIOS disables the circuitry and power-on a per-port basis.	Ports not WOL-enabled are in P2. Ports with WOL are in P0. If all ports are WOL-disabled, internal clocks off.	Ports not WOL-enabled are in P2. Ports with WOL are in P0. If all ports are WOL-disabled, internal clocks off.
USB	The BIOS disables the circuitry and power.	Idle State with internal clocks off.	Idle state with internal clocks off.

9.12 Voltage Identification (VID) Table

Table 9-13. VID Range and Power State Support

VID Range	Required Power State Support
VID greater than 0.5V	PS0, PS1, PS2, PS3
VID = 0.5V	PS0, PS1, PS2, PS3
VID less than 0.5	PS2, PS3

§ §

10 System Address Maps

This chapter describes the four SoC address spaces. They are:

- Physical Address Space, also called the Memory Space
- I/O Space
- PCI Configuration Space

The CPU core only directly accesses the memory space through memory reads and writes, and the I/O space through the IN and OUT instructions.

The PCI configuration space is indirectly accessed through the I/O space and the memory space.

This chapter also describes special registers that are accessible through an SoC internal sideband bus. While not technically an addressing space, these registers are accessed indirectly through index registers in the PCI configuration space. The sideband bus is used externally to set BMBOUND, BMBOUND_HI, relocate the power management firmware to RAM, and initialize other internal registers that need to be setup by the BIOS.

10.1 Physical Address Space Map

The physical address space of 64 GB (36 bits) is used as:

- Memory-Mapped I/O (MMIO) for devices integrated in the SoC.
- DRAM memory implemented as DDR3 SDRAM devices on the platform board.

The CPU core accesses the all 64 GB of physical address space. Integrated devices access their own MMIO registers and DDR3 DRAM.

10.1.1 SoC Transaction Router Memory Map

The SoC transaction router maps the physical address space as follows:

- CPU core to DRAM
- CPU core to I/O device registers mapped to the MMIO memory space
- CPU core to extended PCI registers using the Enhanced Configuration Access Mechanism (ECAM)
- Integrated device (I/O APIC) to CPU cores (local APIC interrupts)

Although 64 GB (36 bits) of physical address space is accessible, some MMIO must exist in 32-bit-address-memory space to allow MMIO access to 32-bit Operating Systems (OS).

The MMIO area is large and is at least 256 MB to provide the ECAM. So as to not waste physical DRAM, the DRAM-access hole created by the address range assigned as MMIO, is re-mapped to memory access requests starting at the 4-GB address. A section DRAM is moved to start at the fixed 4-GB boundary, leaving a hole below 4 GB for MMIO. This creates the following distinct memory regions:

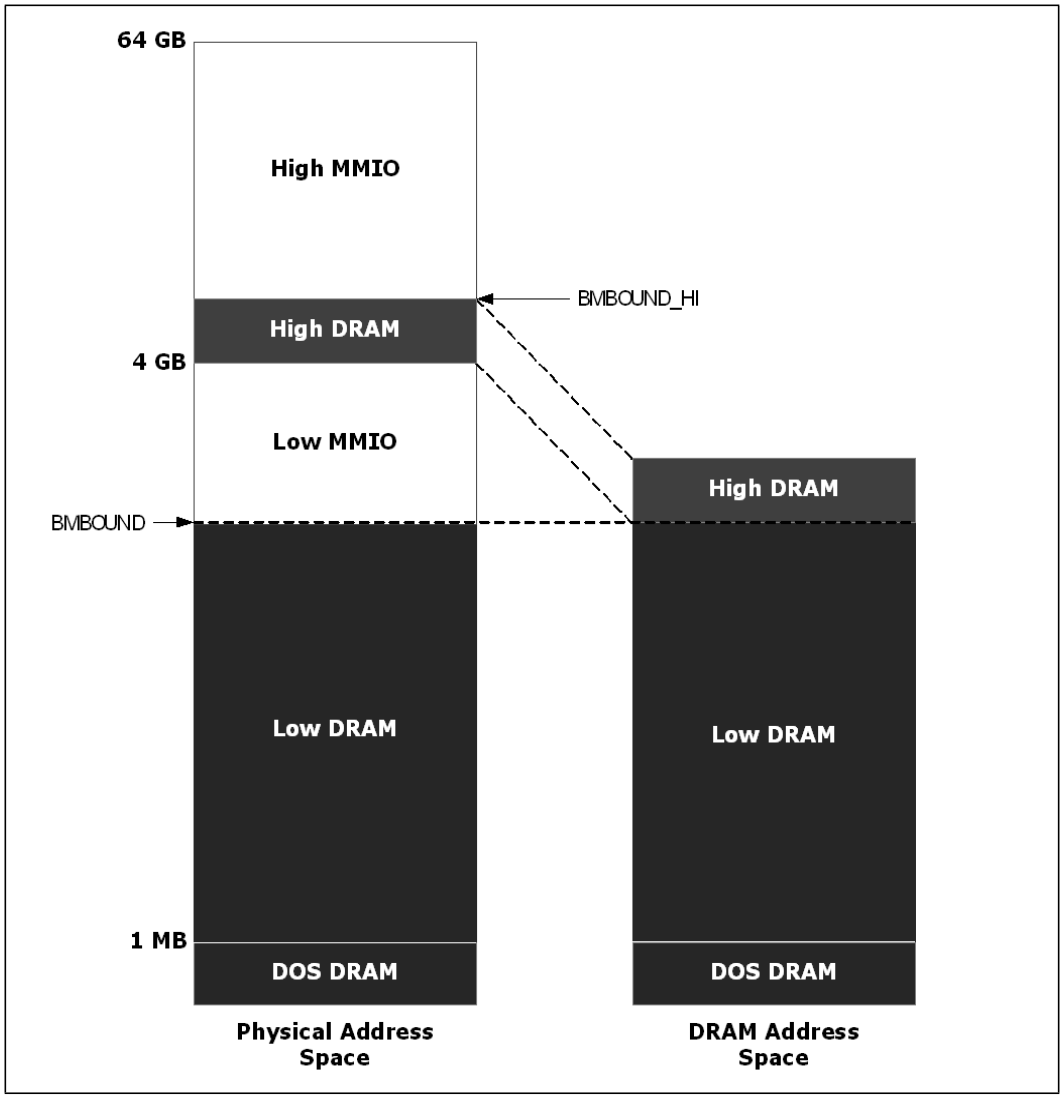
- DOS DRAM + Low DRAM
- Low MMIO
- High DRAM
- High MMIO

Two registers are used to create these regions, BMBOUND and BMBOUND_HI. Their use is shown in Figure 10-1.

BMBOUND and BMBOUND_HI are sideband registers internal to the SoC. They are accessed by the BIOS through the sideband register access mechanism explained in Section 10.4.1, “Sideband Register Access” on page 187.

The values in these two registers must also match those of the 32-bit RTF_BMBOUND and RTF_BMBOUNDHI registers located in the configuration space at bus 0, device 14, function 0, offsets 404h and 408h, respectively.

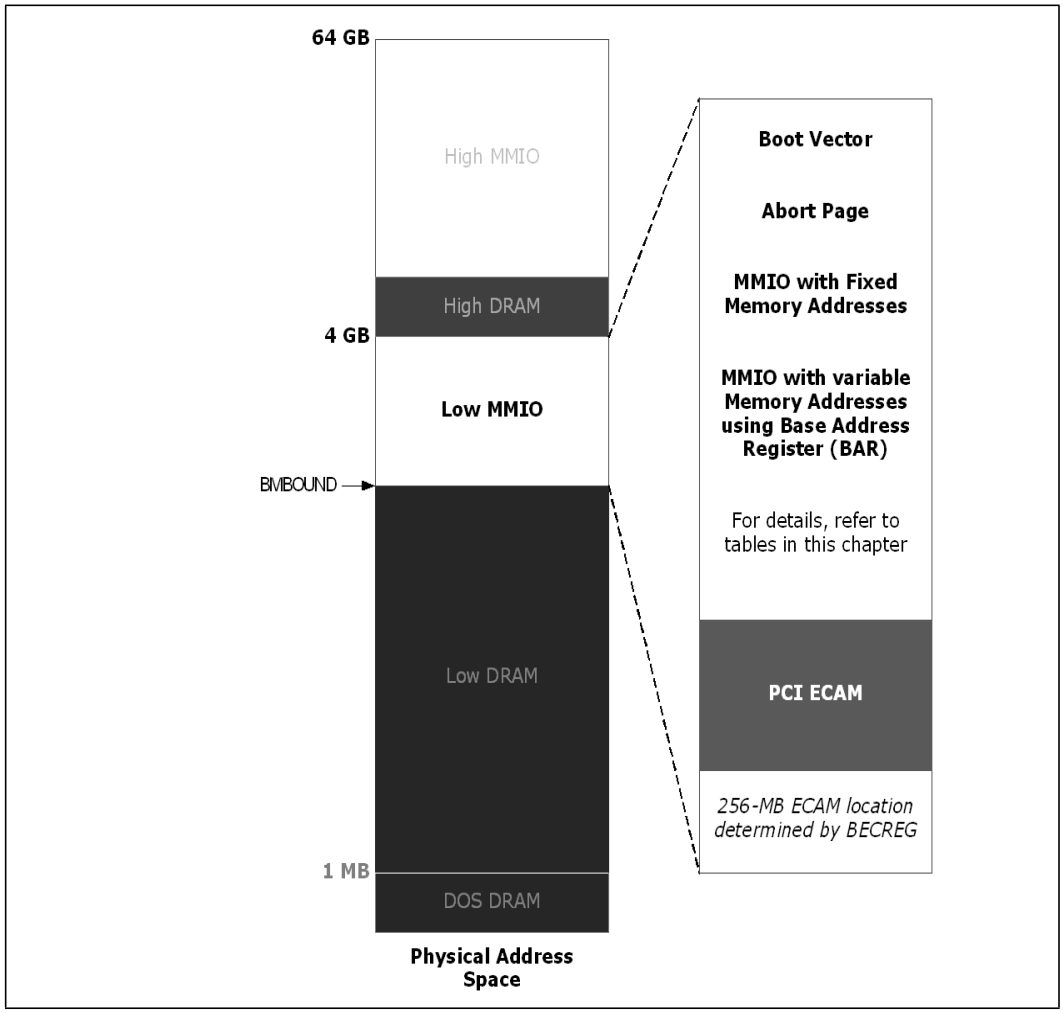
Figure 10-1. Physical Address Space - DRAM and MMIO



10.1.1.1 Low MMIO

The Low MMIO range is shown in Figure 10-2.

Figure 10-2. Physical Address Space - Low MMIO



By default, the CPU core memory reads targeting the Boot Vector range (FFFF_0000h-FFFF_FFFFh) in Low MMIO are sent to the Boot Flash device connected to the platform control unit. Memory write accesses to this area target DRAM. This allows the bootstrap CPU core to fetch boot code from the Boot Flash from either the SoC SPI or LPC interface, and then shadow that code to DRAM. For increased performance, the boot code chooses to reroute read accesses targeting the boot vector range to access DRAM using the BMBOUND.SEND_BOOT_VECTOR_TO_DRAM field. This allows execution of the shadowed boot code from DRAM.

Note: The 16MB BIOS Decode MMIO space under the Boot Vector range is enabled by default via the BDE (BIOS Decode Enable (BDE) register is located in the configuration space at bus 0, device 31 (decimal), function 0, at offset D8h). The BIOS Decode MMIO space allows MMIO accesses to be sent to the LPC bridge or the SPI controller which is determined by the boot selection strap pin FLEX_CLK_SE0.

Upstream writes from the I/O fabric to the **Local APIC** range (FEE0_0000h-FEEF_FFFFh) are sent to the appropriate CPU core APIC.

Write accesses from a CPU core to the **Abort Page** range (FEB0_0000h-FEBF_FFFFh) are dropped, and reads are always return all 1s in binary.

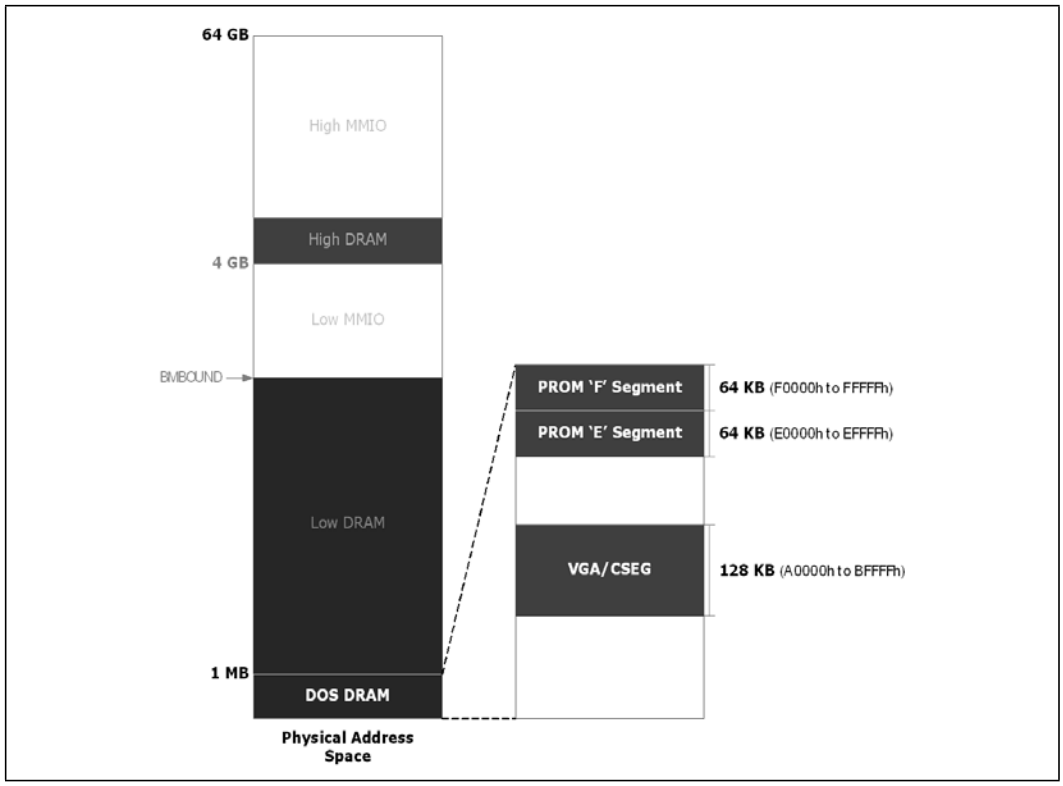
Accesses in the 256-MB **PCI ECAM** range starting at BECREG generate enhanced PCI configuration register accesses when enabled (BECREG.ECENABLE). Unlike traditional memory writes, writes to this range are non-posted when enabled.

Requests to access the Low MMIO range from other sources are sent to the I/O fabric for further address decode based on PCI resource allocations. The I/O fabric subtractive agent for unclaimed accesses is the Platform Controller Unit (PCU).

10.1.1.2 DOS DRAM

The DOS DRAM is the memory space, below 1 MB. In general, accesses from a processor targeting DOS DRAM target system DRAM. Exceptions are shown in Figure 10-3.

Figure 10-3. Physical Address Space - DOS DRAM



Processor writes to the 64 KB (each) **PROM 'E'** and **'F'** segments (000E_0000h-000E_FFFFh and 000F_000h-000F_FFFFh) always target DRAM. The BMISC register directs CPU core reads in these two segments to DRAM or to the I/O fabric (MMIO for the BIOS Decode Enable registers BDE.LEE and BDE.LFE). While accessible to the processor cores, these memory-mapped areas are not accessible to requestors which are PCI Express integrated endpoints, integrated root ports, or the endpoints of the root ports.

Note: The BIOS Decode Enable (BDE) register is located in the configuration space at bus 0, device 31 (decimal), function 0, at offset D8h. This register enables decoding of the E and F segments of memory space and for various other small-address ranges.

The CPU core accesses to the 128 KB **VGA/CSEG** range (000A_0000h-000B_FFFFh) targets DRAM or the I/O fabric (MMIO). The target is selected with the BMISC.ABSEGINDRAM register. The SoC does not support System Management Mode (SMM) code in this range. See Section 10.4, "Sideband Registers" on page 187 for location of BMISC.

The SoC does not support the **ISA Expansion ROM** region (000C_0000h-000D_FFFFh). This area always maps to system DRAM. Access is from the CPU only, no inbound access support. If for some reason an inbound access does occur, this access is aborted.

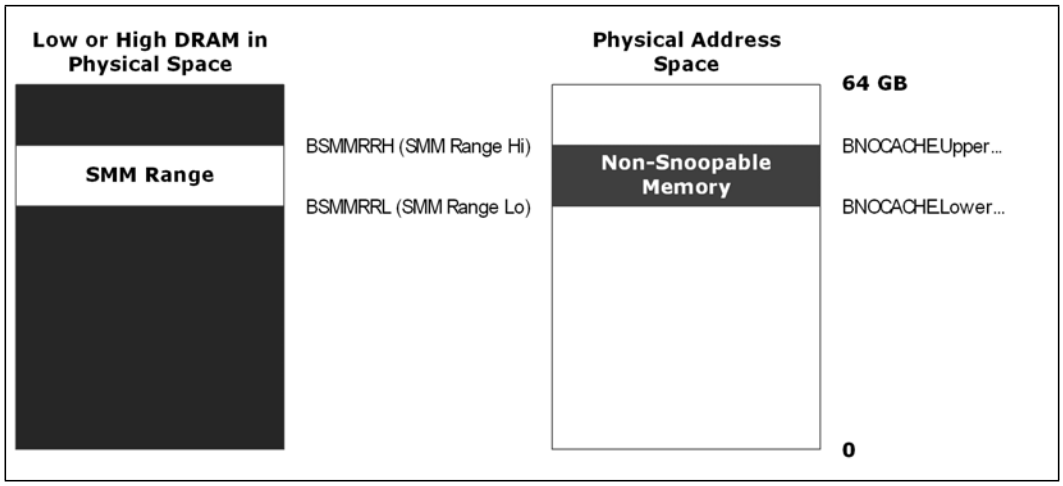
10.1.1.3 Additional Mappings

Two additional mappings are available in the SoC transaction router:

- SMM Range
- Non-Snoop Range

Figure 10-4 shows these mappings which are set by the BIOS and the settings are locked.

Figure 10-4. Physical Address Space - SMM and Non-Snoop Mappings



SMI handlers running on a CPU core execute out of SMM memory. To protect this memory from non-CPU core access, the **SMM Range** (BSMMRRL-BSMMRRH) is programmed anywhere in low or high DRAM space (1 MB aligned). This range only allows accesses from the CPU cores.

To prevent snoops of the CPU cores when DMA devices access a specific memory region, the **Non-Snoopable Memory range** (BNOCACHE.Lower Bound-BNOCACHE.Upper Bound) is programmed anywhere in physical address space. This range is enabled via the BNOCACHECTL register enable bit (BNOCACHECTL.Enabled).

10.1.1.4 Isolated Memory Regions

Seven isolated regions of memory are defined and masked to prohibit certain system agents from accessing memory. The registers that define the regions and provide access control settings are BIMR[0-7]L, BIMR[0-7]H, BIMR[0-7]RAC, and BIMR[0-7]WAC. These are internal Sideband Registers. See Section 10.4, “Sideband Registers” on page 187.

10.1.2 I/O Fabric (MMIO) Map

Memory accesses targeting MMIO are routed by the I/O fabric to programmed PCI ranges, or routed to the PCU by default (subtractive agent). Programmed PCI ranges are moved within low or high MMIO, and most are disabled. **Not all devices are mapped to high MMIO.**

Fixed MMIO is claimed by the Platform Control Unit (PCU). The default regions are listed in Table 10-1 and Table 10-2. The variable (movable) ranges are shown in Table 10-3.

Table 10-1. Internal Devices with Fixed MMIO Addresses

Range Name	Start Address	End Address	Comments
ABORT	FEB0_0000	FEBF_FFFF	Abort Page Region
I/O APIC	FEC0_0000	FEC0_0040	I/O APIC space
HPET	FED0_0000	FED0_03FF	High Performance Event Timer
TPM1.2 (LPC)	FED4_0000	FED4_0FFF	TPM1.2
Local APIC	FEE0_0000	FEEF_FFFF	APIC sends MSIs to CPU and INTR ACKs to Legacy Block interrupt controllers.
LPC	FF00_0000	FF0F_FFFF	BDE.E40[1]
LPC	FF10_0000	FF1F_FFFF	BDE.E50
LPC	FF20_0000	FF2F_FFFF	BDE.E60
LPC	FF30_0000	FF3F_FFFF	BDE.E70
LPC/SPI	FF40_0000	FF4F_FFFF	BDE.E40
LPC/SPI	FF50_0000	FF5F_FFFF	BDE.E50
LPC/SPI	FF60_0000	FF6F_FFFF	BDE.E60
LPC/SPI	FF70_0000	FF7F_FFFF	BDE.E70
LPC	FF80_0000	FF87_FFFF	BDE.EC0
LPC	FF88_0000	FF8F_FFFF	BDE.EC8
LPC	FF90_0000	FF97_FFFF	BDE.ED0
LPC	FF98_0000	FF9F_FFFF	BDE.ED8
LPC	FFA0_0000	FFA7_FFFF	BDE.EE0
LPC	FFA8_0000	FFAF_FFFF	BDE.EE8
LPC	FFB0_0000	FFB7_FFFF	BDE.EF0
LPC	FFB8_0000	FFBF_FFFF	BDE.EF8
LPC/SPI	FFC0_0000	FFC7_FFFF	BDE.EC0
LPC/SPI	FFC8_0000	FFCF_FFFF	BDE.EC8
LPC/SPI	FFD0_0000	FFD7_FFFF	BDE.ED0
LPC/SPI	FFD8_0000	FFDF_FFFF	BDE.ED8
LPC/SPI	FFE0_0000	FFE7_FFFF	BDE.EE0
LPC/SPI	FFE8_0000	FFEF_FFFF	BDE.EE8
LPC/SPI	FFF0_0000	FFF7_FFFF	BDE.EF0
LPC/SPI	FFF8_0000	FFFF_FFFF	BDE.EF8

1. BDE is the 16-bit BIOS Decode Enable (PCIE_REG_BIOS_DECODE_EN) register for setting the BIOS enable for various memory-address ranges. This register is accessed in the configuration space through the iLB Device Bus 0, Function 31 (decimal), Function 0, offset 0D8h. This register affects the BIOS decode regardless of whether the BIOS is resident on the SPI or LPC bus interface.

Table 10-2. Other Fixed Memory Ranges

Device	Start Address	End Address	Comments
Low BIOS (Flash Boot)	000E_0000h	000F_FFFFh	Starts 128 KB below 1 MB; Firmware/BIOS
High BIOS/Boot Vector	FFFF_0000h	FFFF_FFFFh	Starts 64 KB below 4 GB; Firmware/BIOS

Table 10-3 shows the integrated PCI devices that claim memory resources in the MMIO space. See each device functional description chapter for details.

Warning: Variable memory ranges must be set to not conflict with other memory ranges. Unpredictable results can happen if the configuration software allows conflicts to occur. The SoC hardware does not check for conflicts.

Table 10-3. Internal Devices with Variable MMIO Addresses (Sheet 1 of 2)

Range Name	Base Address Register	Size Details	Comments
ECAM	Internal B-Unit: BECREG	256 MB in 32-bit space	For accessing local PCIe* Extended Configuration Space through MMIO
PCI Express* Port 1	D1:F0: EXPPTMBAR	128 KB in 32- or 64-bit space	Reserved
PCI Express Port 1	D1:F0: MEMBASE and MEMLIMIT	Variable in 32-bit space	Memory base and limit for PCI Express* Root Port 1
PCI Express Port 1 (prefetchable)	D1:F0: {PFBASEU, PFBASE} and {PFLIMITU, PFLIMIT}	Variable in 32- or 64-bit space	Prefetchable memory base and limit for PCI Express Root Port 1
PCI Express Port 2	D2:F0: EXPPTMBAR	128 KB in 32- or 64-bit space	Reserved
PCI Express Port 2	D2:F0: MEMBASE and MEMLIMIT	Variable in 32-bit space	Memory base and limit for PCI Express Root Port 2
PCI Express Port 2 (prefetchable)	D2:F0: {PFBASEU, PFBASE} and {PFLIMITU, PFLIMIT}	Variable in 32- or 64-bit space	Prefetchable memory base and limit for PCI Express Root Port 2
PCI Express Port 3	D3:F0: EXPPTMBAR	128 KB in 32- or 64-bit space	Reserved
PCI Express Port 3	D3:F0: MEMBASE and MEMLIMIT	Variable in 32-bit space	Memory base and limit for PCI Express Root Port 3
PCI Express Port 3 (prefetchable)	D3:F0: {PFBASEU, PFBASE} and {PFLIMITU, PFLIMIT}	Variable in 32- or 64-bit space	Prefetchable memory base and limit for PCI Express Root Port 3
PCI Express Port 4	D4:F0: EXPPTMBAR	128 KB in 32- or 64-bit space	Reserved
PCI Express Port 4	D4:F0: MEMBASE and MEMLIMIT	Variable in 32-bit space	Memory base and limit for PCI Express Root Port 4
PCI Express Port 4 (prefetchable)	D4:F0: {PFBASEU, PFBASE} and {PFLIMITU, PFLIMIT}	Variable in 32- or 64-bit space	Prefetchable memory base and limit for PCI Express Root Port 4

Table 10-3. Internal Devices with Variable MMIO Addresses (Sheet 2 of 2)

Range Name	Base Address Register	Size Details	Comments
M/S SMBus	D19:F0: SMTBAR	1 KB in 32- or 64-bit space	SMBus 2.0 master/slave controller (Root Complex Integrated Endpoint)
GbE	D20:Fx: {BAR1, BAR0}	128 KB in 32- or 64-bit space	GbE controller, independent per function
GbE	D20:Fx: BAR2	32 bytes in 32-bit space	GbE controller, independent per function
GbE	D20:Fx: {BAR5, BAR4}	16 KB in 32- or 64-bit space	GbE MSI-X, independent per function
USB2	D22:F0: MBAR	1 KB in 32-bit space	USB 2.0 controller
SATA2	D23:F0: ABAR (BAR5)	2 KB in 32-bit space	SATA2 controller (AHCI base address)
SATA3	D24:F0: ABAR (BAR5)	2 KB in 32-bit space	SATA3 controller (AHCI base address)
ILB	D31:F0: IBASE	512 bytes in 32-bit space	ILB memory space (in PCU)
Proxy Access to P-Unit	D31:F0:PU_BASE	2 KB in 32-bit space	Memory proxy for P-Unit as opposed to HOST uses DW addressing
Proxy Access to I/O Controller	D31:F0: IOBASE	8 KB in 32-bit space	GPIO memory space
Proxy Access to M-PHYs	D31:F0: MPBASE	1 MB in 32-bit space	Memory proxy for SATA, PCIe Root Port, and USB 2.0 PHY interfaces
PMC	D31:F0: PBASE	512 bytes in 32-bit space	PMC memory space (in PCU)
PCU and Proxy Access	D31:F0: RCBA	1 KB in 32-bit space	RCRB memory space (in PCU) and USB 2.0 bridge
SPI	D31:F0: SBASE	512 bytes in 32-bit space	SPI memory space (in PCU)
PCU SMBus	D31:F3: MBARL	32 bytes in 32-bit space	SMBus memory space (in PCU)
Available for 64-bit MMIO Registers	BMBOUND_HI and RTF_BMBOUNDHI	Through the remaining addressable 64 GB	High MMIO for 64-bit MMIO

10.2 I/O Address Space

There are 64 KB + 3 bytes of I/O space (0h-10002h) for accessing I/O registers in the I/O space. Most I/O registers exists for legacy functions in the PCU or for the integrated PCI devices, while some are claimed by the SoC transaction router to allow for external access to the PCI configuration space registers.

10.2.1 SoC Transaction Router I/O Map

The SoC claims I/O transactions for the two 32-bit registers at port CF8h and CFCh used by the software to access the PCI configuration space.

10.2.2 I/O Fabric I/O Map

10.2.2.1 PCU Fixed I/O Address Ranges

Table 10-4 shows the fixed I/O space ranges seen by a processor.

Table 10-4. Fixed I/O Map (Sheet 1 of 2)

Start	End	Target for I/O Reads	Target for I/O Writes	Disable
20h	21h	8259 PIC	8259 PIC	No
24h	25h	8259 PIC	8259 PIC	No
28h	29h	8259 PIC	8259 PIC	No
2Ch	2Dh	8259 PIC	8259 PIC	No
30h	31h	8259 PIC	8259 PIC	No
34h	35h	8259 PIC	8259 PIC	No
38h	39h	8259 PIC	8259 PIC	No
3Ch	3Dh	8259 PIC	8259 PIC	No
40h	42h	8254 PIT	8254 PIT	No
43h	43h	None	8254 PIT	No
50h	53h	8254 PIT	8254 PIT	No
60h	60h	PS/2 Legacy Keyboard/ Mouse Control	PS/2 Legacy Keyboard/Mouse Control	No
61h	61h	NMI Controller	NMI Controller	No
63h	63h	NMI Controller	NMI Controller	Yes, alias to 61h
64h	64h	PS/2 Legacy Keyboard/ Mouse Control	PS/2 Legacy Keyboard/Mouse Control	No
65h	65h	NMI Controller	NMI Controller	Yes, alias to 61h
67h	67h	NMI Controller	NMI Controller	Yes, alias to 61h
70h	70h	None	NMI and RTC	No
71h	71h	RTC	RTC	No
72h	72h	RTC	NMI and RTC	Yes, w/ 73h
73h	73h	RTC	RTC	Yes, w/ 72h,
74h	74h	RTC	NMI and RTC	No
75h	75h	RTC	RTC	No

Table 10-4. Fixed I/O Map (Sheet 2 of 2)

Start	End	Target for I/O Reads	Target for I/O Writes	Disable
76h	76h	RTC	NMI and RTC	No
77h	77h	RTC	RTC	No
80h	8Fh	POST Code registers	POST Code registers	No
92h	92h	INIT (in PMC)	INIT (in PMC)	
A0h	A1h	8259 PIC	8259 PIC	No
A4h	A5h	8259 PIC	8259 PIC	No
A8h	A9h	8259 PIC	8259 PIC	No
ACh	ADh	8259 PIC	8259 PIC	No
B0h	B1h	8259 PIC	8259 PIC	No
B2h	B3h	Power Management (PMC SMI)	Power Management (PMC SMI)	No
B4h	B5h	8259 PIC	8259 PIC	No
B8h	B9h	8259 PIC	8259 PIC	No
BCh	BDh	8259 PIC	8259 PIC	No
1F0h	1F8	Hard Disk Controller	Hard Disk Controller	
2F8h	2FFh	COM2 I/O Space (UART1)	COM2 I/O Space (UART1)	
3B0h	3BBh	PCIe* bridge with BCTL.VGAE set	PCIe bridge with BCTL.VGAE set	Yes, if VGAE is clear
3C0h	3DFh	PCIe bridge with BCTL.VGAE set	PCIe bridge with BCTL.VGAE set	Yes, if VGAE is clear
3F8h	3FFh	COM1 I/O Space (UART0)	COM1 I/O Space (UART0)	
4D0h	4D1h	8259 PIC	8259 PIC	No
CF8h *Dword (32-bit) access only*	CFBh	Access to PCI configuration space. Also needed to access internal sideband registers.	Access to PCI configuration space. Also needed to access internal sideband registers.	No
CF9h	CF9h	Reset Generator	Reset Generator	No
CFCh	CFFh	Access to PCI configuration space. Also needed to access internal sideband registers.	Access to PCI Configuration space. Also needed to access internal sideband registers.	No

10.2.2.2 Variable I/O Address Ranges

Table 10-5 shows the variable I/O decode ranges. These blocks are independently configured and enabled by the platform BIOS. They are set using base address registers (BARs) or other similar means. Plug-and-Play (PnP) software (PCI/ACPI) uses their configuration mechanisms to set and adjust these values.

Warning: The variable I/O ranges are not set to conflict with other I/O ranges. Unpredictable results can happen if the configuration software allows conflicts to occur. The hardware does not check for conflicts.

Table 10-5. Variable I/O Map (Sheet 1 of 2)

Range Name	Mappable		Size (Bytes)	Target
PCI Express* Port 1	D1:F0: IOBASE	Anywhere in 64K I/O space	Up to IOLIMIT	PCIe* Legacy I/O Access
PCI Express Port 2	D2:F0: IOBASE	Anywhere in 64K I/O space	Up to IOLIMIT	PCIe Legacy I/O Access
PCI Express Port 3	D3:F0: IOBASE	Anywhere in 64K I/O space	Up to IOLIMIT	PCIe Legacy I/O Access
PCI Express Port 4	D4:F0: IOBASE	Anywhere in 64K I/O space	Up to IOLIMIT	PCIe Legacy I/O Access
GbE	D20:Fx: BAR2	Anywhere in 64K I/O space	32	GbE Independent per function
SATA2	D23:F0: PCMDBA (BAR0)	Anywhere in 64K I/O space	8	SATA2 Primary Command Block
SATA2	D23:F0: PCTLBA (BAR1)	Anywhere in 64K I/O space	4	SATA2 Primary Control Block
SATA2	D23:F0: SCMDBA (BAR2)	Anywhere in 64K I/O space	8	SATA2 Secondary Command Block
SATA2	D23:F0: SCTLBA (BAR3)	Anywhere in 64K I/O space	4	SATA2 Secondary Control Block
SATA2	D23:F0: LBAR (BAR4) [IDE mode]	Anywhere in 64K I/O space	16	SATA2 Legacy IDE Base Address
SATA2	D23:F0: LBAR (BAR4) [not IDE]	Anywhere in 64K I/O space	32	SATA2 AHCI Index Data Pair Base Address
SATA2	D23:F0: SIDPBA (BAR5) [IDE mode]	Anywhere in 64K I/O space	16	SATA2 Index Data Pair Base Address
SATA3	D23:F0: PCMDBA (BAR0)	Anywhere in 64K I/O space	8	SATA3 Primary Command Block
SATA3	D24:F0: PCTLBA (BAR1)	Anywhere in 64K I/O space	4	SATA3 Primary Control Block
SATA3	D24:F0: SCMDBA (BAR2)	Anywhere in 64K I/O space	8	SATA3 Secondary Command Block

Table 10-5. Variable I/O Map (Sheet 2 of 2)

Range Name	Mappable		Size (Bytes)	Target
SATA3	D24:F0: SCTLBA (BAR3)	Anywhere in 64K I/O space	4	SATA3 Secondary Control Block
SATA3	D24:F0: LBAR (BAR4) [IDE mode]	Anywhere in 64K I/O space	16	SATA3 Legacy IDE Base Address
SATA3	D24:F0: LBAR (BAR4) [not IDE]	Anywhere in 64K I/O space	32	SATA3 AHCI Index Data Pair Base Address
SATA3	D24:F0: SIDPBA (BAR5) [IDE mode]	Anywhere in 64K I/O space	16	SATA3 Index Data Pair Base Address
ACPI Includes TCO (WDT)	D31:F0: ABASE	Anywhere in 64K I/O space	128	Power management: ACPI and TCO (in PMC)
GPIO Proxy access	D31:F0: GBASE	Anywhere in 64K I/O space	256	GPIO I/O space
SMBus	D31:F3: IOBAR	Anywhere in 64K I/O space	32	SMBus (in PCU)

10.3 PCI Configuration Space

All PCI devices/functions are shown in Table 10-6.

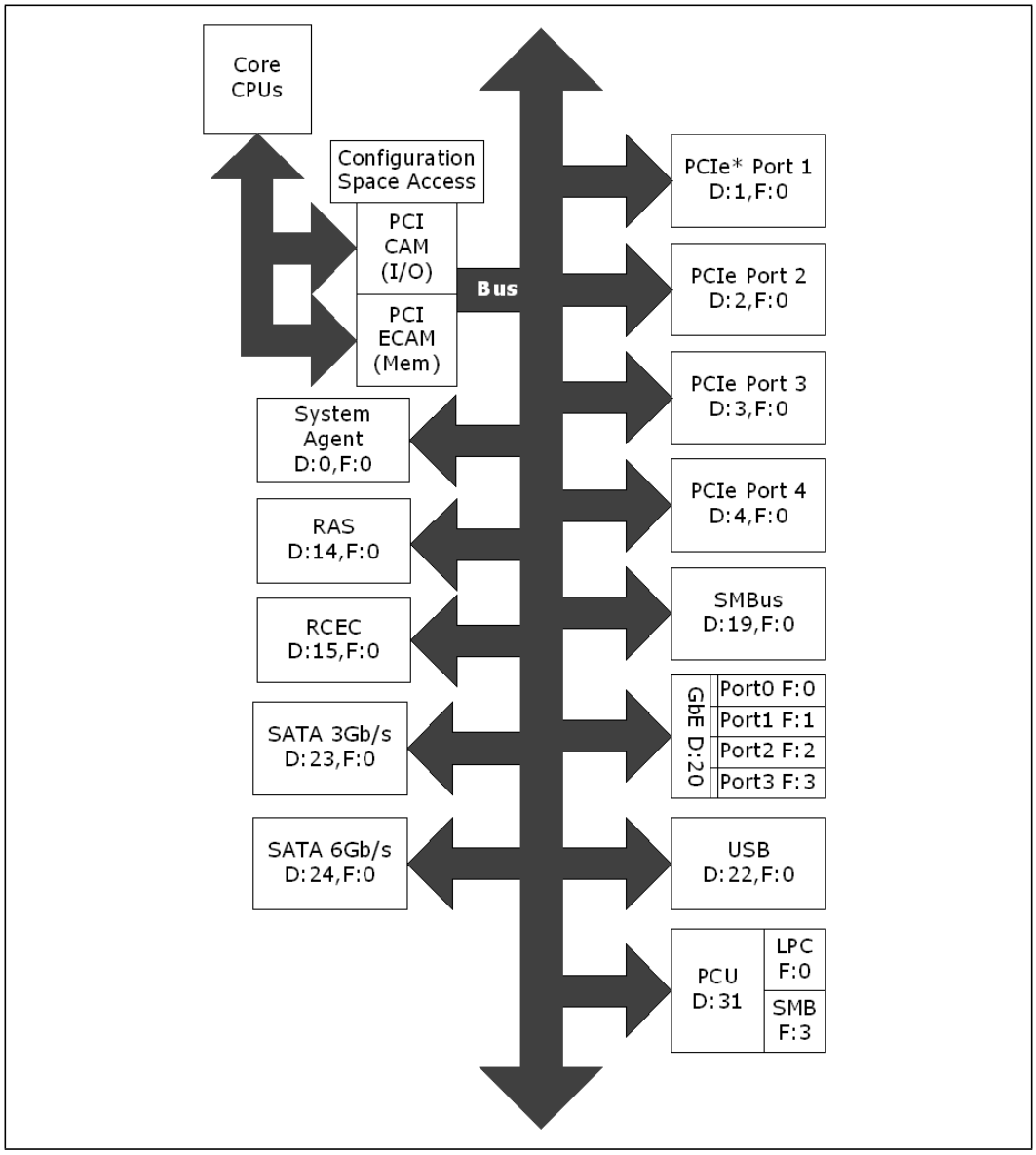
Table 10-6. PCI Devices and Functions (Sheet 1 of 2)

Bus	Device (decimal)	Function	Device ID	Description	Comment
0	0	0	0x1F00	SoC Transaction Router	Internal SoC Fabric
			0x1F01		Internal SoC Fabric
			0x1F02		Internal SoC Fabric
			0x1F03		Internal SoC Fabric
			0x1F04		Internal SoC Fabric
			0x1F05		Internal SoC Fabric
			0x1F06		Internal SoC Fabric
			0x1F07		Internal SoC Fabric
			0x1F08		Internal SoC Fabric
			0x1F09		Internal SoC Fabric
			0x1F0A		Internal SoC Fabric
			0x1F0B		Internal SoC Fabric
			0x1F0C		Internal SoC Fabric
			0x1F0D		Internal SoC Fabric
			0x1F0E		Internal SoC Fabric
			0x1F0F		Internal SoC Fabric
0	1	0	0x1F10	PCI Express* Root Port 1	
0	2	0	0x1F11	PCI Express Root Port 2	
0	3	0	0x1F12	PCI Express Root Port 3	
0	4	0	0x1F13	PCI Express Root Port 4	
0	14	0	0x1F14	Reliability, Availability and Serviceability (RAS)	
0	15	0	0x1F16	Root Complex Event Collector (RCEC)	
0	19	0	0x1F15	SMBus 2.0	
0	20	0-3	0x1F40	GbE - 1000BASE-KX	Configured for use with backplane
			0x1F41	GbE - Serial Gigabit Media Independent Interface (SGMII)	Configured for use with external PHY component
			0x1F45	GbE - 2.5 GbE	Configured for use with backplane
0	22	0	0x1F2C	USB 2.0	

Table 10-6. PCI Devices and Functions (Sheet 2 of 2)

Bus	Device (decimal)	Function	Device ID	Description	Comment
0	23	0	0x1F22	SATA2	The SATA2 Device ID is configured based on a number of straps/fuse/ registers that influence the SKU. 1F20 SoC 4-Port IDE SATA2 Controller 1F21 SoC DE SATA2 Controller 1F22 SoC AHCI SATA2 Controller 1F23 SoC AHCI SATA2 Controller
0	24	0	0x1F32	SATA3	SATA3 Device ID is influenced by the fuse, strap and register setting (MAP). 1F20 SoC 4-Port IDE SATA3 Controller 1F21 SoC IDE SATA3 Controller 1F22 SoC AHCI SATA3 Controller 1F23 SoC AHCI SATA3 Controller
0	31	0	0x1F38	Platform Controller Unit (PCU)	
			0x1F39		
			0x1F3A		
			0x1F3B		
		3	0x1F3C	PCU SMBus	

Figure 10-5. SoC Device Map



10.4 Sideband Registers

The SoC has a number of internal registers called Sideband Registers that are indirectly accessed through the configuration space. This mechanism provides access to the logic units within the host bridge. These registers are useful in configuring the memory map, power management, and other internal units.

10.4.1 Sideband Register Access

The internal sideband registers are accessed indirectly by writing/reading the Message Control Register (MCR), the Sideband Data Register (MDR) and the Sideband Packet Extension Register (MCRE) that are located at bus 0, device 0, function 0, offsets D0h, D4h, and D8h, respectively. See Table 10-7.

Table 10-7. Sideband Register Access Registers

Sideband Register Access Register	Name	Number of Bits	Bus 0, Device 0, Function 0 Offset
Message Control Register (MCR)	CUNIT_MSG_CTRL_REG	32	D0h
Message Data Register (MDR)	CUNIT_MSG_DATA_REG	32	D4h
Message Control Register Extension (MCRE)	CUNIT_MSG_CTRL_REG_EXT	32	D8h

10.4.1.1 Sideband Registers for Address Mapping

An abbreviated list of sideband registers mentioned in this chapter are shown in Table 10-8. The sideband registers are typically accessed by the BIOS.

Table 10-8. Sideband Registers Mentioned in This Chapter

Port ID (B-Unit)	Register Offset	Register Name	Description
P03h	23h	BNOCACHE	Non-Cached Region
	24h	BNOCACHECTL	Non-Cached Region Control
	25h	BMBOUND	Memory and I/O Boundary Register
	26h	BMBOUND_HI	Memory and I/O HI Boundary Register
	27h	BECREG	Extended Configuration Space Configuration Register
	28h	BMISC	Miscellaneous Configuration Register
	2Eh	BSMMRRL	System Management Range Register - Low
	2Fh	BSMMRRH	System Management Range Register - High
	80h	BIMR0L	Isolated Memory Region 0 Low
	81h	BIMR0H	Isolated Memory Region 0 High
	82h	BIMR0RAC	Isolated Memory Region 0 Read Access Control
	83h	BIMR0WAC	Isolated Memory Region 0 Write Access Control

Note: Only Isolated Memory Region 0 is shown in the table for BIMR[0-7]L, BIMR[0-7]H, BIMR[0-7]RAC, and BIMR[0-7]WAC. There are a total of 7 regions.

§ §

11 Gigabit Ethernet (GbE) Controller

11.1 Introduction

The Gigabit Ethernet (GbE) controller is a PCI device with PCI Express* architecture capabilities integrated in the SoC. The controller provides an interface that supports four independent gigabit Ethernet Media Access Control Ports (MACs). Each MAC has SGMII, SerDes, 1000BASE-KX, and 2.5-GbE configuration capability. Routing the interfaces either directly to a backplane or connecting to an external SGMII-capable Physical-Layer device (PHY) enables 1000BASE-T connections.

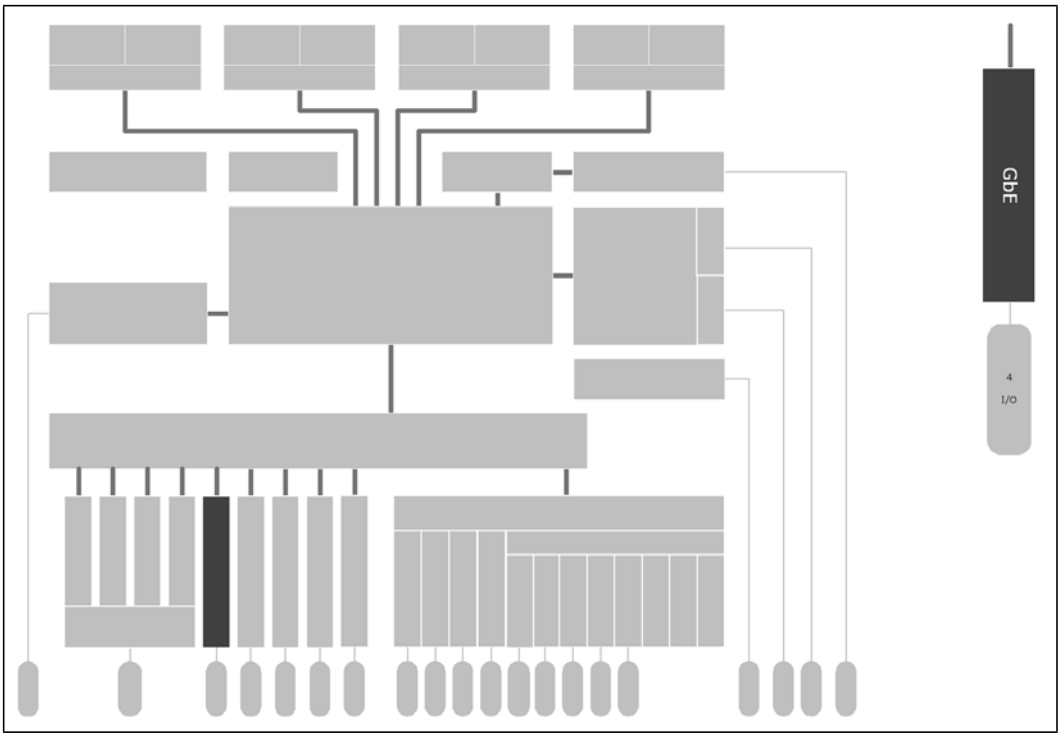
Note: An MDIO/MDC connection is required for SGMII support. Refer to Appendix D of the *Intel® Atom™ Processor C2000 Product Family Platform Design Guide (PDG)* for guidance on which PHYs are supported.

The integrated GbE controller provides an interface for system management. The interface can be configured to operate as a Network Controller Sideband Interface (NC-SI) or as an SMBus interface. The GbE controller can provide the clock for each interface.

The GbE controller also provides LED control, programmable pins, and an SPI interface for the controller EEPROM.

Note: SoC customers are advised to contact Intel concerning their GbE EEPROM needs. Intel provides a number of EEPROM image files for SoC customers to use in their designs. These files are considered "starter images" and to some extent can be altered for a customer's particular usage.

Figure 11-1. GbE Interface Covered in This Chapter



11.2 Programmer's Reference Manual

Much of the programming-related information and register descriptions for the integrated Gigabit Ethernet controller are in a separated document. Refer to the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*, document 537426.

This document provides information needed by platform board hardware designers and provides an overview of the registers. The descriptions of the MMIO registers and MSI-X registers located in memory space, the registers in I/O space, and the contents of the EEPROM are described in the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*.

11.3 Feature List

A complete list of features is shown in the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*. A partial list is shown in here.

- Four network interface ports (LAN Port 0, 1, 2, and 3) each with its own MAC.
- PCI Express* integrated endpoint with one device (20 decimal) with four functions (0, 1, 2, and 3).
- Virtual LAN (VLAN) support. 64 or more per LAN Port depending on OS.
- Four-wire SPI interface for the controller's EEPROM.
- Four LED driver outputs which can be used as GPIOs if not used.
- Two software-defined pins.
- Management Data Input/Output (MDIO) for external SGMII-capable Physical-Layer (PHY) device
- LAN function disable capability.
- Magic packet wake-up enable with unique MAC address.
- ACPI register set and power-down functionality supporting D0 and D3 states.
- Full wake-up support (APM and ACPI 2.0).
- Smart power down at S0 no link and Sx no link.
- Support for SMBus or NCSI to connect BMC.

11.4 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The signal/pin name
- **Direction**: The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type found in Chapter 31, "Signal Names and Descriptions"
- **Description**: A brief explanation of the signal function

Table 11-1. Signals (Sheet 1 of 3)

Signal Name	Direction/ Type	Description
GBE_TXP[3:0]	O	**SerDes/SGMII Serial Data Output Port**: Differential SGMII/ SerDes transmit interface.
GBE_TXN[3:0]	O	**SerDes/SGMII Serial Data Output Port**: Differential SGMII/ SerDes transmit interface.
GBE_RXP[3:0]	I	**SerDes/SGMII Serial Data Input Port**: Differential SGMII/ SerDes receive interface.
GBE_RXN[3:0]	I	**SerDes/SGMII Serial Data Input Port**: Differential SGMII/ SerDes receive interface.
GBE_REFCLKP GBE_REFCLKN	I	**GbE 100-MHz differential clock** with 100 ppm maximum jitter. External SerDes/SGMII differential 100-MHz reference clock from an external generator. This clock must be powered from the Suspend (SUS) power well. When the device is enabled for the 2.5-GbE operation, the standard 100-MHz reference clock must be replaced with a 125-MHz reference clock.
GBE_OBSP GBE_OBSN	O	**Observability port**. In normal operation, configure as GBE_RCOMP.
GBE_EE_DI/GPIO_SUS13	O	**GbE EEPROM Data Input:** Data is output to EEPROM. If all four LAN Ports are disabled via soft straps, this signal can be used as GPIO SUS Port 13.
GBE_EE_DO/GPIO_SUS14	I	**GbE EEPROM Data Output**: Data is input from EEPROM. If all four LAN Ports are disabled via soft straps, this signal can be used as GPIO SUS Port 14.
GBE_EE_SK/GPIO_SUS15	O	**GbE EEPROM Serial Clock:** Serial clock output to EEPROM Operates at ~2 MHz. If all four LAN Ports are disabled via soft straps, this signal can be used as GPIO SUS Port 15.
GBE_EE_CS_N/ GPIO_SUS16	O	**GbE EEPROM Chip Select**: Chip select Output to EEPROM. If all four LAN Ports are disabled via soft straps, this signal can be used as GPIO SUS Port 16.
GBE_SMBD/NCSI_TX_EN	I/O, OD	**GbE SMBus Clock**: One clock pulse is generated for each data bit transferred. An external pull- down of 10K resistor is required. Resistor value should be calculated based on the bus load. (Refer to the Platform Design Guide.) If the GBE_SMBD interface is not used, the signals can be used as NCSI_TX_EN transmit enable (input). ***Note:*** If not used, should have an external pull-down resistor.
GBE_SMBCLK/ NCSI_CLK_IN	I/O, OD	**GbE SMBus Clock**: One clock pulse is generated for each data bit transferred. An external pull-up resistor is required. If the SMBus interface is not used, the signals can be used as the NCSI_CLK_IN signal. As an input signal, the NCSI_CLK_IN must be connected to the 50-MHz NC-SI REF_CLK generator on the platform board. This same signal pin can be programmed to provide the 50-MHz NC-SI REF_CLK for the NC-SI devices on the platform board including the SoC. If so programmed, the NCSI_CLK_IN pin also functions as the "NCSI_CLK_OUT" of the SoC. ***Note:*** If this pin is not used, it must be connected to an external pull-down resistor.

Table 11-1. Signals (Sheet 2 of 3)

Signal Name	Direction/ Type	Description
GBE_SMBALRT_N/ NCSI_CRS_DV	I/O, OD	**GbE SMBus Alert:** Acts as an interrupt of a slave device on the SMBus. An external pull-up resistor is required. If the GBE_SMBALRT_N interface is not used, the signals can be used as NCSI_CRS_DV Carrier Sense/Receive Data Valid (CRS/ DV).
GBE_SDP0_0/GPIO_SUS17	I/O	**GbE Port 0 SW Defined Pin 0:** The SDP pins are reserved pins that are software programmable with write/read, input/output capability. These default to inputs upon power up but may have their direction and output values defined in the EEPROM. The SDP bits may be mapped to the General Purpose Interrupt bits when configured as inputs. SDP can be used for IEEE* 1588 standard interface. The SDP0_0 pin can be used as a watchdog output indication. If the GBE_SDP0_0 interface is not used, the signal can be used as GPIO SUS Port 17.
GBE_SDP0_1/ GPIO_SUS18/ NCSI_ARB_IN	I/O	**GbE Port 0 SW Defined Pin 1:** The SDP pins are reserved pins that are software programmable with write/read, input/output capability. These default to inputs upon power up, but may have their direction and output values defined in the EEPROM. The SDP bits may be mapped to the General Purpose Interrupt bits when configured as inputs. SDP can be used for IEEE 1588 standard interface. The SDP0_1 pin can be used as a watchdog output indication. SFP sideband signals are not supported. If GBE_SDP0_1 interface is not used, the signal can be used as GPIO SUS Port 18. If none of the above functions are used, the signal can be used as NCSI_ARB_IN arbitration input.
GBE_LED0/GPIO_SUS19	O	**GBE_LED[3:0]** Programming: 0000: Port 0 link up 0001: Port 1 link up 0010: Port 2 link up 0011: Port 3 link up 0100: Port 0 activity 0101: Port 1 activity 0110: Port 2 activity 0111: Port 3 activity 1000: Ports 0-3 ink up 1001: Ports 0-1 link up 1010: Ports 0-3 activity 1011: Ports 0-1 activity If the GBE_LED[3:0] interface is not used, the signals can be used as GPIO SUS Port [22:19].
GBE_LED1/GPIO_SUS20	O	
GBE_LED2/GPIO_SUS21	O	
GBE_LED3/GPIO_SUS22	O	
NCSI_RXD1/GPIO_SUS23	O	**NCSI Receive Data 1**: Data signals to the Manageability Controller (MC). Pin strap to enable GbE in S5 (1'b1 to enable – the default). If the NCSI_RXD1 signal is not used, the signal can be used as GPIO SUS Port 23. Refer to Section 16.2, "Pin-Based (Hard) Straps" on page 357.
GBE_MDIO0_I2C_CLK/ GPIO_SUS24	I/O, OD	**Gigabit Ethernet Controller Management Channel 0 Clock (out)**: Serial clock for the management channel. Can also be configured as an I^2C (OD) clock. If the GBE_MDIO0_I2C_CLK interface is not used, the signal can be used as GPIO SUS Port 24.
GBE_MDIO0_I2C_DATA/ GPIO_SUS25	I/O, OD	**Gigabit Ethernet Controller Management Channel 0 Data (T/S)**: Serial data for the management channel. Can also be configured as I^2C (OD) data. If the GBE_MDIO1_I2C_DATA interface is not used, the signal can be used as GPIO SUS Port 25.
GBE_MDIO1_I2C_CLK/ GPIO_SUS26/NCSI_TXD1	I/O, OD	**Gigabit Ethernet Controller Management Channel 1 Clock (out)**: Serial clock for the management channel. Can also be configured as an I^2C (OD) clock. If the GBE_MDIO1_I2C_CLK interface is not used, the signal can be used as GPIO SUS Port 26. If none of the above functions are used, the signal can be used as NCSI_TXD1 Transmit Data 1. Data signals from the MC. ***Note:*** If not used, should have an external pull-up resistor.

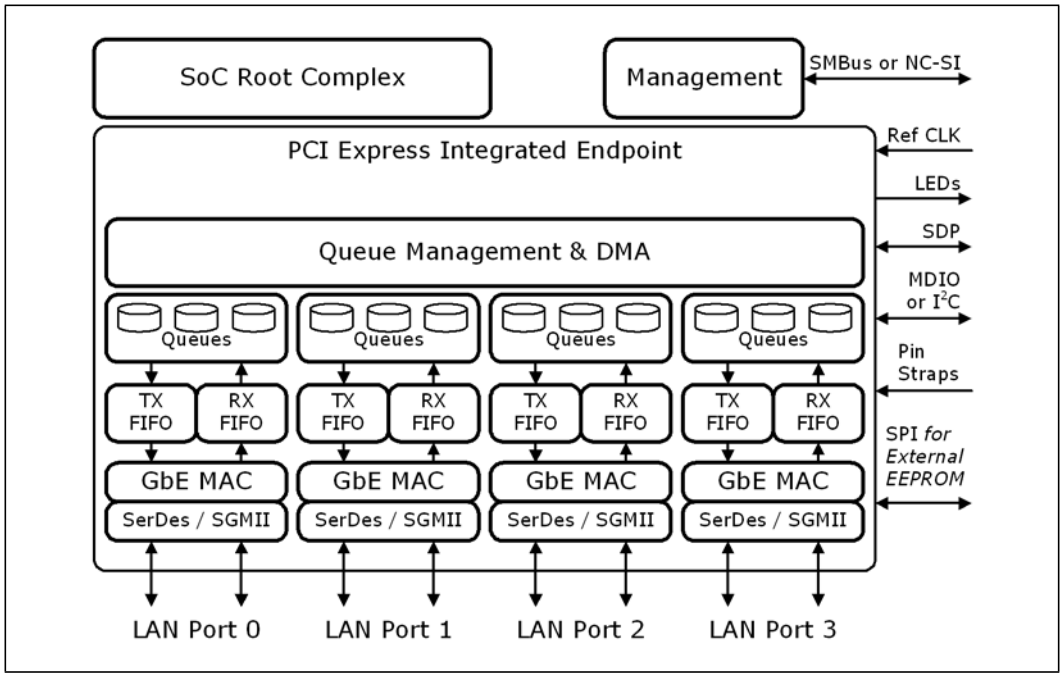
Table 11-1. Signals (Sheet 3 of 3)

Signal Name	Direction/ Type	Description
GBE_MDIO1_I2C_DATA/ GPIO_SUS27/NCSI_TXD0	I/O, OD	**Gigabit Ethernet Controller Management Channel 1 Data (T/S)**: Serial data for the management channel. Can also be configured as I^2C (OD) data. If the GBE_MDIO1_I2C_DATA interface is not used, the signal can be used as GPIO SUS Port 27. If none of the above functions are used, the signal can be used as NCSI_TXD0 Transmit Data 0. Data signals from the MC. ***Note:*** If not used, should have an external pull-up resistor.
GPIO_SUS1/NCSI_RXD0	I/O	**SUS Well GPIO_1:** General purpose Customer I/O. If GPIO_SUS1 is not used, the signal can be used as NCSI_RXD0 Receive Data 0 signal to the MC. This pin is also a pin-strap input. If sensed low, the 2.5-GbE capability, if available, is disabled. This pin must be sampled high for the 2.5-GbE capability to function. This pin is temporarily pulled-down internally during the sample period. An external pull-up resistor is needed during the sample period to enable 2.5 GbE. Refer to Section 16.2, "Pin-Based (Hard) Straps" on page 357.
STRAP_NCSI_EN/ Y59_RSVD/ NCSI_ARB_OUT	O	**NCSI_ARB_OUT**: NC-SI hardware arbitration token output pin. ***Note:*** This pin is also a hard pin-strap. When it is a logic high at power-up, it indicates the NC-SI interface is to be used rather than the GBE_SMBus. Refer to Section 16.2, "Pin-Based (Hard) Straps" on page 357.

11.5 Architectural Overview

The integrated GbE controller uses descriptor queues and Direct Memory Access (DMA). Figure 11-2 shows an architectural overview of the controller and the interface to the platform board. The number of LAN Ports available varies by product SKU. Key parts of the diagram are explained. The queues and the DMA are described in the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*.

Figure 11-2. System Architecture and Interface



11.5.1 PCIe* Integrated Endpoint

From a systems view, the GbE controller is a PCIe integrated endpoint consisting of one device with four functions. The device number is 20 decimal and its four function numbers are 0, 1, 2, and 3.

The controller is highly configurable. The device ID for each of the four functions are assigned, and its capabilities are enabled/disabled by the data contained in the controller's EEPROM device. The configurable PCI and PCIe capabilities, and their offsets from the start of each function location in PCI configuration space are in Table 11-2.

Table 11-2. PCI and PCIe Capabilities Supported

Offset in Configuration Space (hexadecimal)	Type of Capability	Capability
40	PCI	PCI Power Management Interface (PMI)
50	PCI	Message Signaled Interrupts (MSI)
70	PCI	Message Signaled Interrupts, extended (MSI-X)
A0	PCI	PCI Express* Capability
E0	PCI	Vital Product Data (VPD)
100	PCI Express* Extended	Advanced Error Reporting (AER)
140	PCI Express* Extended	Device Serial Number
1D0	PCI Express* Extended	Access Control Services (ACS)

The PCI Device IDs that can be assigned to a function via the EEPROM data:

- 0x1F40 - 1000BASE-KX (1 GbE) interface to backplane.
- 0x1F41 - Serial Gigabit Media Independent Interface (SGMII). Requires SGMII-compliant PHY component. For guidance see a list in the appendix of the *Intel® Atom™ Processor C2000 Product Family Platform Design Guide (PDG)*.
- 0x1F45 - 2500BASE-X (2.5 GbE) interface to backplane.

Note: The "Dummy Function" Device ID that is available on some Intel LAN components is not supported by the SoC.

For information of the LAN interface and supported standards, see Section 11.5.6, "LAN Port Interface" on page 199.

Each of the four functions has three Base Address Registers (BARs) that are assigned address values during PCI bus enumeration. See Table 11-3.

Table 11-3. Base Address Registers

Offset in Configuration Space (hexadecimal)	32-bit Register Name	Capability	Data bytes accessible in given space
10	BASE_ADDR_0	64-bit BAR for Memory Space for Memory-Mapped I/Os (MMIOs)	128 K
14	BASE_ADDR_1		
18	BASE_ADDR_2	32-bit BAR for I/O Space (for access to the GbE controller's internal command and status registers)	32
1C	BASE_ADDR_3	*Reserved*	
20	BASE_ADDR_4	64-bit BAR for Memory Space for Message Signaled Interrupts, extended (MSI-X)	16 K
24	BASE_ADDR_5		

There are two device-specific registers located in configuration space for each function. They provide one of the two possible methods for accessing the GbE controller's internal command and status registers. This first method uses PCI configuration space of each of the four functions:

- 98h - IOADDR (32 bits)
- 9Ch - IODATA (32 bits)

This access through configuration space is not available if the function is assigned the Dummy Function device ID.

The other method uses the system's I/O space. Once the BASE_ADDR_2 BAR is enumerated for a particular function, it can be used to access the IOADDR and IODATA registers:

- BASE_ADDR_2 (I/O base address) plus 0 - IOADDR (32 bits)
- BASE_ADDR_2 (I/O base address) plus 4 - IODATA (32 bits)

Note: There are many configuration options that affect the parameter assignments and operation of the integrated GbE controller. This document only provides an overview and does not show all possibilities available to the SoC customer. For detailed descriptions, see the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*.

11.5.2 Setting Up PCI Device Presence and Non-Presence

PCI Device and Function presence or non-presence must be established before PCI enumeration and before the integrated GbE controller is released from PCI Reset. Soft Straps for the GbE controller are provided for this purpose.

11.5.2.1 Soft Straps for GbE Controller

For a given product SKU, the SoC Soft Strap settings are the first setup parameters applied to the SoC devices. The Soft Strap settings are applied to the SoC devices while the platform and integrated PCI devices are still reset. In this reset state, the integrated GbE controller maps PCI function 0 to LAN Port 0, function 1 to LAN Port 1, etc.

There are four SoC Soft Straps that affect the four LAN Ports. By default, all LAN Ports (and PCI functions) of a particular product SKU are enabled. LAN Ports (and possibly their corresponding PCI function) 1, 2, and 3 can be disabled by setting its Soft Strap to a 1. The GBE_ALL_DISABLE Soft Strap, if set, overrides the settings of the three Soft Straps for LAN Ports 1, 2, and 3. When set, the GBE_ALL_DISABLE Soft Strap also disables LAN Port 0. See Table 16-5, “Flash Descriptor Soft Strap” on page 362.

When set to a “1,” the GBE_ALL_DISABLE Soft Strap, the GbE controller is not enumerated, and so, does not exist. In this case, and if the system-management capabilities of the integrated GbE controller are not wanted, the GbE EEPROM is not needed in the platform design.

Support in S5 for Wake-on-LAN and/or Manageability requires setting the ‘GbE powered in S5’ bit in the Flash Descriptor soft straps. With this soft strap set the GbE MAC will retain connection through type 1 and 3 system resets.

Note: If the GbE Controller is configured to provide the 50-MHz NC-SI “REF_CLK” signal to the rest of the platform board via the NCSI_CLK_IN signal (pin P50, an output as well as an input), then the GBE_ALL_Disable Soft Strap must be a “0” (at least Function 0 of the device is enabled).

11.5.3 Disabling LAN Ports and PCI Functions by EEPROM

After the Soft Strap settings are applied (Soft Strap GBE_ALL_DISABLE must be "0" to get to this point) and after the integrated PCI devices exit the reset state, the information in the EEPROM is used to setup the configuration registers accessed during PCI enumeration. The EEPROM has the following control bits in the Software Defined Pins Control words. There is one word for each of the four LAN Ports at EEPROM LAN-word offset 20h:

- LAN_DIS
- LAN_PCI_DIS

If the LAN Port is enabled by its Soft Strap setting, the LAN_DIS and LAN_PCI_DIS settings in EEPROM are applied to LAN Ports 1, 2, and 3. A Soft-Strap enabled LAN Port 0 cannot be disabled by its LAN_DIS and LAN_PCI_DIS bits in EEPROM. LAN Ports 1, 2, and 3 and associated PCI functions both enabled unless:

LAN_DIS = 1: The LAN is disabled. Here both PCIe function and LAN access for manageability are disabled.

LAN_PCI_DIS = 1: The associated LAN PCI function is disabled and is not enumerated and thus not connected to the host as an integrated PCIe endpoint. Even so, the LAN Port's MAC is kept active and fully functional for manageability purposes and for BMC pass-through traffic.

11.5.4 Disabling PCI Functions by BIOS

The SoC BIOS can disable the integrated GbE controller PCI functions. The memory-mapped Function Disable (FUNC_DIS) register of the Power Management Controller (PMC) portion of the SoC Platform Controller Unit (PCU) contains four bits to disable the four PCI functions.

11.5.5 Mapping PCI Functions to LAN Ports

The LAN Ports 0, 1, 2, and 3 are mapped to the four PCI functions 0, 1, 2, and 3 respectively. There are no other mapping options supported by the SoC.

11.5.6 LAN Port Interface

The GbE controller LAN Port interface provides a complete CSMA/CD function supporting IEEE* 802.3 1000BASE-KX and 2500BASE-X (2.5 GbE) implementations. The controller also supports external PHY components that comply with the Serial Gigabit Media Independent Interface (SGMII). Refer to Appendix D of the *Intel® Atom™ Processor C2000 Product Family Platform Design Guide (PDG)* for guidance on which PHYs are supported.

The LAN Ports, each with its own MAC and set of transmit and receive queues, performs all of the functions required for transmission, reception, and collision handling called out in the standards.

Each LAN Port MAC can each be configured to use a different media interface. Selection of the media interface is programmable the 2-bit LINK_MODE field (bits 22, 23) of the MAC Extended Device Control (CTRL_EXT) register, located at offset 18h of the MMIO of the PCI function mapped to the LAN Port. The default link mode is set via the 2-bit Link Mode field (bits 4, 5) of the Initialization Control 3, located at 16-bit word offset 24h of the particular LAN-Port base address of the EEPROM. The link modes are shown in Table 11-4.

Table 11-4. LAN Port Link Mode

Bits 5:4 - Initialization Control 3 (EEPROM) also Bits 23:22 - CTRL_EX (MMIO)	LAN Port Interface Link Mode
00	*Reserved*
01[1]	1000BASE-KX (1 GbE) 2500BASE-X (2.5 GbE)
10	SGMII
11	*Reserved*

1. The MAC for each LAN Port functions as 1 GbE with a 100-MHz reference clock or 2.5 GbE with a 125-MHz reference clock.

The internal MAC and PCS supports 10/100/1000/2500 Mb/s operation. With SGMII link mode, both half- and full-duplex operation are supported at 10/100 Mb/s and only full-duplex operation at other SGMII speeds. With 1000BASE-KX or 2500BASE-KX link mode, only full-duplex operation is supported.

The 1000BASE-KX (1 Gb/s) and 2500BASE-X (2.5 Gb/s) link mode is used for Ethernet-over-backplane implementations. In this mode, only parallel detection is supported and the LAN Port does not support the full Auto-Negotiation for Backplane Ethernet protocol as defined in IEEE Standard 802.3-2008 Clause 73, Auto-Negotiation for Backplane Ethernet.

The 2500BASE-X link mode is a special, enhanced speed mode of the 1000BASE-X link mode. The SoC has a pin strap to allow 2500BASE-X operation (see Section 11.5.8, “Pin Straps” on page 201) and the GbE reference clock of the SoC must be driven with a 125-MHz differential signal instead of a 100-MHz signal (see Section 11.5.7, “Reference Clock Input” on page 201). Also, the appropriate PCI Device ID (see Section 11.5.1, “PCIe* Integrated Endpoint” on page 195) must be set for the PCI function mapped to the LAN Port.

The SGMII link mode is used when external PHY components are used. For proper network operation, both the LAN Port MAC and the external PHY component must be properly configured, either explicitly via software or via hardware auto-negotiation, with identical speed and duplex settings. The SGMII Auto-Negotiation functionality is similar to what is defined in Clause 37 of the IEEE Standard 802.3-2008. The GbE controller provides an external MDIO/MDC interface to configure external PHYs connected to the LAN Port SGMII interface. See Section 11.5.12, "MDIO and I^{2}C Interface" on page 204.

Refer to Appendix D of the *Intel® Atom™ Processor C2000 Product Family Platform Design Guide (PDG)* for guidance on which external PHYs are supported.

Each GbE controller LAN Port can provide MAC loopback where the controller's internal serial/deserial unit is not functional and data sent by the SoC root complex is fed-back to the root complex. Each LAN port also has the loopback capability where the serial/deserial unit is functional and the data normally sent over the LAN Port interface to the backplane or external PHY is instead fed-back to the serial/deserial unit and back to the root complex.

For detailed information on configuring, auto negotiation, Ethernet flow control, loopback and controlling the LAN Ports and external PHYs, see the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*. Also refer to the PRM for these LAN Port features:

- Device power management and power states
- DMA Coalescing
- Broadcast Wake Up
- IPv4/IPv6 packet-detect support
- Magic Packet detection and operation
- Packet pattern flexible filters

Electrical and timing specifications are in Section 33.3, "2.5 and 1 Gigabit Ethernet (GbE) Interface" on page 642. Board design guidelines are given in the *Intel® Atom™ Processor C2000 Product Family Platform Design Guide (PDG)*.

11.5.7 Reference Clock Input

The integrated GbE controller requires a differential, non spread-spectrum, reference clock for most of its MAC operation. The platform board must provide this clock. The center frequency of the reference clock must be 100 MHz, except when 2500BASE-X (2.5 GbE) is used. For 2500BASE-X the center frequency must be 125 MHz. The electrical and timing requirements are in Section 33.16.2, "GbE Reference Clock" on page 670.

11.5.8 Pin Straps

There are three SoC hard pin-straps that are related to the integrated GbE controller and need attention. Three of the SoC pins are:

- Pin W54 - GPIO_SUS1: Used to enable/disable the 2.5 GbE feature.
- Pin Y59 - NCSI_ARB_OUT: Used to choose SMBus or NC-SI as the interface for management.
- Pin V63 - NCSI_RXD1: Whether or not the GbE controller has power during the S5 (Soft Off) Sleep State.

Additional information for these pins is in Section 16.2, "Pin-Based (Hard) Straps" on page 357.

11.5.9 LED Interface

The GbE controller provides four output drivers intended for driving external LED circuits. Each of the four LED outputs can be individually configured to select the particular event, state, or activity, which is indicated on that output. In addition, each LED can be individually configured for output polarity as well as for blinking versus non-blinking (steady-state) indication.

The configuration for LED outputs is specified via the read/write LED Control (LEDCTL) Register, located in Memory-Mapped I/O (MMIO) at offset E00h. LEDCTL controls the setup of the internal signals routed to the external LEDs according to the GbE LEDs Mux Control (LEDS_MUX_CTRL) Register located in (MMIO) at offset 8130h.

Furthermore, the hardware-default configuration for all the LED outputs, can be specified via EEPROM fields, thereby supporting LED displays configurable to a particular OEM preference. There are two 16-bit words in EEPROM for each LAN Port 0, 1, 2, and 3:

- LED 0,2 Configuration Defaults - 16-bit Word offset 1Fh
- LED 1,3 Configuration Defaults - 16-bit Word offset 1Ch

The offset values mentioned above are from the EEPROM 16-bit word base address for a particular LAN Port. The four base address values, 0h, 80h, C0h, and 100h, are shown in Table 11-9.

Using the LEDCTL register in MMIO, each of the four LEDs can be configured to use one of a variety of sources for output indication. The "Mode" fields of the LEDCTL register control the LED source. The "Invert" bits allow the LED source to be inverted before being output to the LED or observed by the blink-control logic. LED outputs are assumed to normally be connected to the negative side (cathode) of an external LED. The LEDCTL "Blink" BLINK bits control whether the LED should be blinked (on for 200 ms, then off for 200 ms) while the LED source is asserted. The blink control might be especially useful for ensuring that certain events, such as ACTIVITY indication, cause LED transitions that are visible to the human eye.

Note: When LED Blink mode is enabled, the appropriate LED Invert bit should be set to 0b. The LINK/ACTIVITY source functions slightly different from the others when BLINK is enabled. The LED is off if there is no LINK, on if there is LINK and no ACTIVITY, and blinking if there is LINK and ACTIVITY.

The dynamic LED modes (FILTER_ACTIVITY, LINK/ACTIVITY, COLLISION, ACTIVITY, PAUSED) should be used with LED Blink mode enabled.

For addition information concerning the LEDCTL register, LEDS_MUX_CTRL register, the EEPROM registers, and LED control, refer to the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*.

Electrical and timing specifications are in Section 33.7.2, "GbE LED and Software-Defined Pins (SDP)" on page 656. Board design guidelines are given in the *Intel® Atom™ Processor C2000 Product Family Platform Design Guide (PDG)*.

11.5.10 Software-Defined Pins

The GbE controller provides two Software-Defined Pins (SDP) for IEEE 1588 auxiliary device connections and other miscellaneous hardware- or software-control purposes. These pins, and their functions, are bound to a specific LAN device. The pins can be individually configured to act as standard inputs, General-Purpose Interrupt (GPI) inputs, or output pins.

The use, direction, and values of SDP pins are controlled and accessed in each PCI function's Memory-Mapped I/O (MMIO) using fields in the Device Control (CTRL) register (MMIO offset 0, and aliased at offset 4h) and the 32-bit Extended Device Control (CTRL_EXT) register (MMIO offset 18h).

The internal SDP ports are routed to the SoC pins based on the programmable GbE SDPs Mux Control (SDPS_MUX_CTRL) register located at MMIO offset 8134h for each PCI function.

The default direction of each of the pins is configurable via the EEPROM as well as the default value of any pins configured as outputs. To avoid signal contention, all pins are set as input pins until after the EEPROM configuration has been loaded by the GbE controller. Each of the four LAN Ports has a Software Defined Pins Control register located at 16-bit word offset 20h from the LAN Ports base address in EEPROM. The four base address values, 0h, 80h, C0h, and 100h, are shown in Table 11-9.

In addition to all pins being individually configurable as inputs or outputs, they can be configured for use as General-Purpose Interrupt (GPI) inputs. To act as GPI pins, the desired pins must be configured as inputs. A separate GPI interrupt-detection enable is then used to enable rising-edge detection of the input pin (rising-edge detection occurs by comparing values sampled at the internal clock rate as opposed to an edge-detection circuit). When detected, a corresponding GPI interrupt is indicated in the Interrupt Cause Read (ICR) register located at MMIO offset 1500h for each PCI function.

For addition information concerning the CTRL register, CTRL_EXT register, the EEPROM registers, and SDP control and operation, refer to the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*.

Electrical and timing specifications are in Section 33.7.2, "GbE LED and Software-Defined Pins (SDP)" on page 656. Board design guidelines are given in the *Intel® Atom™ Processor C2000 Product Family Platform Design Guide (PDG)*.

11.5.11 SPI Interface

Unless the SoC Soft Strap GBE_ALL_DISABLE is "1," an external EEPROM device must be connected to the four-wire Serial Peripheral Interface (SPI) bus interface of the GbE controller. The controller provides a 2-MHz (typical) serial clock for the bus. Electrical and timing specifications are in Section 33.6, "Network Controller EEPROM Interface" on page 654. EEPROM product recommendations and board design guidelines are given in the *Intel® Atom™ Processor C2000 Product Family Platform Design Guide (PDG)*.

11.5.12 MDIO and I^{2}C Interface

When the LAN Port interface is set to operate in SGMII mode, an external PHY device can be accessed by the GbE controller through one of two PHY-management interfaces types:

- an MII Management interface where communication is through the read/write MDI Control (MDIC) register in MMIO, offset 20h.
- or a two-wire standard-mode I^{2}C interface where communication is through the read/write SGMII I^{2}C Command (I2CCMD) register in MMIO, offset 1028h.

The MII Management interface is described in the subsection titled "Management functions" of Clause 22 "Reconciliation Sublayer (RS) and Media Independent Interface (MII)" of the *IEEE Standard 802.3*-2008*. The GbE controller supports the optional Clause 45 electrical characteristics described in the specification but it does not support the logical extensions of Clause 45.

The SoC has two MDIO/I2C interface ports:

- MDIO0 - associated with LAN Port 0. If it is not configured to be dedicated to LAN Port 0, this MDIO can be shared with LAN Ports 1, 2, and 3.
- MDIO1 - associated with LAN Port 1. This MDIO cannot be shared with the other LAN Ports.

More is said about sharing in Section 11.5.12.1, "Sharing the MDIO0 Interface" on page 205.

For each of these two interface ports, the SoC provides a receiver circuit and an open-drain driver as the interface to the platform board. Electrically, either the I^{2}C or MDIO interface can be implemented on the platform board.

Regardless which of the two interface types are used by the design, the following bits must both be set for the interface pins to function (see Table 11-5):

- I^{2}C Enabled - bit 25 of the Extended Device Control (CTRL_EXT) register in MMIO, offset 18h
- Destination - bit 31 of the MDC/MDIO Configuration (MDICNFG) register in MMIO, offset E04h. Bit 31 is initialized by the value of External MDIO (bit 2) of the LAN Port's EEPROM Initialization Control 3 word.

CTRL_EXT and MDICNFG registers exist for LAN Port 0 (MDIO0) and for LAN Port 1 (MDIO1).

Table 11-5. Enabling MDIO/I²C Interface Pins

Destination Bit	I²C Enabled Bit		Registers that provide Functionality
0	0	→	pins not functional
0	1	→	pins not functional
1	0	→	pins not functional
1	1	→	MDIC, I2CCMD

11.5.12.1 Sharing the MDIO0 Interface

MDIO0, if enabled, can be configured as either "shared" or "not-shared" (a.k.a. "dedicated") through the value of the COM_MDIO bit of the MDC/MDIO Configuration (MDICNFG) register, which is an MMIO register for LAN Port 0. Regardless of its COM_MDIO setting, MDIO0 is always used by LAN Port 0.

COM_MDIO = 0: Not Shared

COM_MDIO = 1: Shared

See Table 11-6.

For LAN Port 1, MDIO1, if enabled, is used unless the COM_MDIO bit for both LAN Port 0 and LAN Port 1 are "1" indicating "shared." If both are "1," MDIO0 is used if it is enabled.

For LAN Port 2 and for LAN Port 3, when their COM_MDIO bit indicates "shared," MDIO0 is used by the LAN Port. When the bit indicates "not-shared," no MDIO interface is used by the particular LAN Port.

Table 11-6. MDIO Interface for LAN Ports

COM_MDIO Bit Value for each LAN Port (LP) MDICNFG register				Shared MDIO (MDIO0) Interface Mode	MDIO Interface that applies to LAN Port (LP) Interface			
LP0	LP1	LP2	LP3		LP0	LP1	LP2	LP3
0	X[1]	X	X	MDIO0 is not shared →	MDIO0	MDIO1	none	none
1	0	0	0	MDIO0 is shared if needed →	MDIO0	MDIO1	none	none
1	0	0	1	MDIO0 is shared if needed →	MDIO0	MDIO1	none	MDIO0
1	0	1	0	MDIO0 is shared if needed →	MDIO0	MDIO1	MDIO0	none
1	0	1	1	MDIO0 is shared if needed →	MDIO0	MDIO1	MDIO0	MDIO0
1	1	0	0	MDIO0 is shared if needed →	MDIO0	MDIO0	none	none
1	1	0	1	MDIO0 is shared if needed →	MDIO0	MDIO0	none	MDIO0
1	1	1	0	MDIO0 is shared if needed →	MDIO0	MDIO0	MDIO0	none
1	1	1	1	MDIO0 is shared if needed →	MDIO0	MDIO0	MDIO0	MDIO0

1. "X" means "don't care."

COM_MDIO is bit 30 of the MDC/MDIO Configuration (MDICNFG) register, located at MMIO offset E04h for each LAN Port of the Ethernet controller. The preset value of bit 30 is provided by the value of COM_MDIO, bit 3 of the Initialization Control 3 word for each LAN Port in EEPROM (LAN word offset 24h).

For detailed information on the MDC/MDIO interface software controls, see the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*. Also refer to the *Intel® Atom™ Processor C2000 Product Family Supported Ethernet Port Configuration Application Note*, Document number 509576.

Electrical and timing specifications are in Section 33.4, "Network Controller MDIO Interface" on page 652. Board design guidelines are given in the *Intel® Atom™ Processor C2000 Product Family Platform Design Guide (PDG)*.

11.5.13 SMBus and NC-SI Interface

The GbE controller provides two interfaces to connect to an external Baseboard Management Controller (BMC). The board design can use either the System Management Bus (SMBus) interface or the Network Controller Sideband Interface (NC-SI). A SoC pin-based hard strap is used to designate which of the two interfaces is used in the platform board design. See Section 11.5.8, "Pin Straps" on page 201 and Section 16.2, "Pin-Based (Hard) Straps" on page 357.

11.5.13.1 SMBus 2.0

SMBus 2.0 is an optional interface for pass-through and/or configuration traffic between an external BMC and the SoC integrated GbE controller. As an SMBus master, the GbE controller provides the SMBus clock at 84 kHz, nominal. As a target, it functions with an SMBus clock between 10 kHz and 100 kHz provided by the bus master. Electrical and timing specifications are in Section 33.7.1, "GbE SMBus 2.0 Interface" on page 656.

The SMBus channel behavior and the commands used to configure or read status from the SoC integrated GbE controller are outlined in the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*.

The SoC GbE interface also enables reporting and controlling the device using the Management Component Transport Protocol (MCTP) protocol over SMBus. The MCTP interface is used by the BMC to control only the interface, not for pass-through traffic. All network ports are mapped to a single MCTP endpoint on SMBus. For information, refer to the PRM.

11.5.13.2 NC-SI and REF_CLK

The NC-SI is an optional connection to an external BMC defined by the Distributed Management Task Force (DMTF) protocol defined in the *Network Controller Sideband Interface (NC-SI) Specification (DSP0222)*. The It operates as a single interface with an external BMC, where all traffic between the GbE controller and the BMC flows through the interface. The GbE controller interface supports the standard DMTF NC-SI protocol and supports both pass-through traffic between the BMC and integrated GbE controller LAN functions as well as configuration traffic between the BMC and other SoC internal units as outlined in the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*.

The NC-SI Specification describes a 50-MHz clock REF_CLK that is used by all the NC-SI devices. This clock must be provided by the platform board or one of the NC-SI devices. As an option the SoC integrated GbE controller can provide this 50-MHz REF_CLK. The internal circuitry of the SoC calls this signal NCSI_CLK_OUT which is internally connected to the NCSI_CLK_IN signal pin P50. The Functions Control register in EEPROM at 16-bit word location 21h, contains the NC-SI Output Clock Disable (bit 13). When this bit is 0, the GbE controller drives the 50-MHz NC-SI REF_CLK (NCSI_CLK_OUT) via pin P50. When this bit is 1, the GbE controller does not drive the NC-SI clock and pin P50 is an input, NCSI_CLK_IN. The default setting of bit 13 is 0. If this bit is 0 (NCSI_CLK_OUT enabled), the Power Down Enable (bit 15) of Device Rev ID in EEPROM word 1Eh must also be 0 (its default setting).

Note: If the GbE Controller is configured to provide the 50-MHz NC-SI clock signal to the rest of the platform board, then the SoC Soft Strap GBE_ALL_Disable must be a "0" (device is enabled).

Note: If NC-SI is not used, then Functions Control register bit 13 must be set (NCSI_CLK_OUT disabled).

Functions Control register bits 15 and 14 control the drive strength of the GbE controller's NC-SI Clock (NCSI_CLK_OUT) Pad Drive Strength and NC-SI Data (NCSI_CRS_DV and NCSI_RXD[1:0]) Pad Drive Strength respectively.

The Multi-Drop NC-SI (bit 11) of the Common Firmware Parameters 2 located in the Firmware section of the EEPROM defines the NC-SI topology as Point-to-Point (bit 11 = 1) or Multi-Drop (bit 11 = 0, the default setting).

The GbE controller dynamically drives its NCSI_CRS_DV and NCSI_RXD[1:0] output signals as required by the sideband protocol:

- On power-up, the SoC floats the NC-SI outputs except for NCSI_CLK_OUT.
- If the GbE controller operates in Point-to-Point topology mode, it starts driving the NC-SI outputs some time following power-up.
- If the GbE controller operates in a Multi-Drop topology mode, it drives the NC-SI outputs as configured by the BMC.

Additional information is available in the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*.

Electrical and timing specifications are in Section 33.5, "Network Controller Sideband Interface (NC-SI)" on page 653.

Board design guidelines for both SMBus and NC-SI are given in the *Intel® Atom™ Processor C2000 Product Family Platform Design Guide (PDG)*.

11.6 EEPROM

The integrated GbE controller requires a single, external, EEPROM device to configure parameters for all LAN Ports and PCI Functions including MAC Addresses, LED behaviors, receive packet filters for manageability, wakeup capability. EEPROM-less operation is not supported.

Contents of the EEPROM are always addressed as 16-bit words. When an offset value is shown, it is always in terms of 16-bit words.

Some of the EEPROM 16-bit words are used to specify hardware parameters that affect all four GbE interfaces such as those affecting circuit behavior. Other EEPROM words are associated with a specific GbE interface. All GbE interfaces access the EEPROM to obtain their respective configuration settings.

11.6.1 EEPROM Starter Images

Customers are advised to contact Intel concerning their GbE EEPROM needs. Intel provides a number of EEPROM image files for SoC customers to use in their designs. These files are considered “starter images” and to some extent can be altered for a customer’s particular usage. Intel provides tools for customers to program images to the system for various Operating Systems. Contact the Intel sales representative for details. If Intel needs to provide EEPROM updates to customers, Intel provides the necessary software tools to do this.

The nine EEPROM starter images currently offered (list is subject to change) are shown in Table 11-7. Each image is padded with bits to provide an image of 8K 16-bit words (16K bytes).

Table 11-7. EEPROM Starter Images

EEPROM Image	Size (16-bit Words)[1]	1000BASE-KX (1 Gb/s)	2500BASE-X (2 Gb/s	SGMII	Management using SMBus	Management using NC-SI
1	8K	X			*no management*	
2	8K	X			X	
3	8K	X				X
4	8K		X		*no management*	
5	8K		X		X	
6	8K		X			X
7	8K			X	*no management*	
8	8K			X	X	
9	8K			X		X

1. The text refers to EEPROM data and addressing in terms of 16-bit words. 8K words equals 16K bytes.

The size of the EEPROM device used in the board design is expressed in the four-bit EEPROM Size field, bits [13:10], of the EEPROM Sizing and Protected Fields located at address 12h of the EEPROM. See Table 11-8. The starter image files all have this set to the default value of 16K bytes (8K words).

Table 11-8. EEPROM Size Field

Bits [13:10] of EEPROM 16-bit Word 12h	EEPROM Size (Number of Bytes)	EEPROM Size (Number of 16-bit Words)	Note
0000 0001 through 0110	*Reserved*	–	
0111	16 K	8 K	Default
1000	32 K	16 K	
1001	64 K	32 K	
1010 1011 through 1111	*Reserved*	–	

11.6.2 EEPROM Map

The EEPROM is divided into five major regions and a number of sub-regions as shown in Table 11-9. Table 11-10 and Table 11-11 show a list of EEPROM words of customer interest. For detailed descriptions of these words, see the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*. EEPROM words defined for exclusive use by Intel are not shown.

Table 11-9. EEPROM Regions

16-bit Word Location (Hexadecimal)	Number of 16-bit Words (Decimal)	Parameters and Information Area	
000 to 07F	128	000 to 002	MAC Address for LAN Port 0
		003 to 02F	Mixture of LAN Port 0 and Words used by the controller as a whole.
		030 to 03E	Preboot eXecution Environment (PXE) Area
		03F	Software Checksun for words 000-03F
		040 to 04F	Software Area
		050 to 05F	Intel Firmware Pointers
080 to 0BF	64	LAN Port 1 (with Software Checksum for 080-0BF at 0BF)	
0C0 to 0FF	64	LAN Port 2 (with Software Checksum for 0C0-0FF at 0FF)	
100 to 13F	64	LAN Port 3 (with Software Checksum for 100-13F at 13F)	
140 to *varies*	*varies*	• Intel Firmware Structures • Vital Product Data (VPD) Area • Command/Status Register Configuration • Analog Configuration and Structures	

Table 11-10. EEPROM 16-Bit Word Locations - One Word for GbE Controller (Sheet 1 of 2)

EEPROM 16-Bit Word Address (Hexadecimal)	Number of 16-Bit Words	Name of Word	Customer Parameter?
3	1	Compatibility	Yes
4	1	Port Identification LED Blinking	Yes
5	1	EEPROM Image Revision	Yes
6	2	Available for OEM use	Yes
8	2	PBA[1] Number/Pointer and Block	Yes
A	1	Initialization Control Word 1	
B	1	Subsystem ID	
C	1	Subsystem Vendor ID	Yes
E	1	Vendor ID	Yes
12	1	EEPROM Sizing and Protected Fields	
1E	1	Device Rev ID	Yes
21	1	Functions Control	Yes
2D	1	Start of RO Area	
2F	1	VPD[2] Pointer	
30	1	Setup Options PCI Function 0	
31	1	Configuration Customization Options PCI Function 0	

Table 11-10. EEPROM 16-Bit Word Locations - One Word for GbE Controller (Sheet 2 of 2)

EEPROM 16-Bit Word Address (Hexadecimal)	Number of 16-Bit Words	Name of Word	Customer Parameter?
32	1	PXE[3] Version	
33	1	Flash (Option ROM) Capabilities	
34	1	Setup Options PCI Function 1	
35	1	Configuration Customization Options PCI Function 1	
36	1	iSCSI Option ROM Version	
37	1	Alternate MAC Address Pointer	
38	1	Setup Options PCI Function 2	
39	1	Configuration Customization Options PCI Function 2	
3A	1	Setup Options PCI Function 3	
3B	1	Configuration Customization Options PCI Function 3	
3D	1	iSCSI Boot Configuration Pointer	
40	2	Reserved	
42	2	Image Unique ID	
44	12	*Reserved*	
50	1	*Reserved*	
52	46	*Reserved*	
140	varies	Option ROM and Firmware Structures	

1. Printed Board Assembly
2. Vital Product Data
3. Preboot eXecution Environment

Table 11-11. EEPROM 16-Bit Word Locations - One Word for Each LAN Port (Sheet 1 of 2)

EEPROM 16-Bit Word Address (Hexadecimal)				Number of 16-Bit Words	Name of Word
LAN Port 0	LAN Port 1	LAN Port 2	LAN Port 3		
0	80	C0	100	3	Ethernet Address
D	8D	CD	10D	1	Device ID
F	8F	CF	10F	1	Initialization Control Word 2
10	90	D0	110	1	*Reserved* (must be = 0xFFFF)
11	91	D1	111	1	Management Pass-Through LAN Configuration Pointer
13	93	D3	113	1	Initialization Control Word 4
16	96	D6	116	1	MSI-X Configuration
17	97	D7	117	1	Software Reset CSR Auto Configuration Pointer
18	98	D8	118	1	*Reserved*
1C	9C	DC	11C	1	LED 1,3 Configuration Defaults
1F	9F	DF	11F	1	LED 0,2 Configuration Defaults
20	A0	E0	120	1	Software Defined Pins Control

Table 11-11. EEPROM 16-Bit Word Locations - One Word for Each LAN Port (Sheet 2 of 2)

EEPROM 16-Bit Word Address (Hexadecimal)				Number of 16-Bit Words	Name of Word
LAN Port 0	LAN Port 1	LAN Port 2	LAN Port 3		
24	A4	E4	124	1	Initialization Control 3
27	A7	E7	127	1	CSR Auto Configuration Power-Up Pointer
3F	BF	FF	13F	1	Checksum Word[1] *for word offset 00h through 3Fh of the particular LAN Port EEPROM base address.*

1. Only the Checksum Word is Software Accessible. The other words are used by the GbE Controller hardware to configure the subsystem.

11.6.3 Unique MAC Address

The SoC customer assigns their own unique six-byte MAC Address for each LAN Port. The EEPROM has three 16-bit word locations defined for each of the four LAN Ports. The "Ethernet Address" three-word locations are shown in Table 11-11. The six-byte MAC Address is programmed to the EEPROM as shown in this example, where the desired MAC Address for LAN Port 2 is 00-A0-C9-00-00-03:

Word C0h = A000h
Word C1h = 00C9h
Word C2h = 0300h

11.6.4 Read EEPROM Contents

When installed on a board, the contents of the EEPROM are not accessible to the customer software in the GbE Controller Memory-Mapped I/O (MMIO). Intel does provide a software tool for customers to view the EEPROM contents. Contact the local Intel sales representative. EEPROM contents are also visible in registers loaded into driver space.

11.6.5 Autoload from EEPROM and Resets

The GbE Controller hardware initializes and configures the controller using the contents of the EEPROM when various state and reset conditions occur. These conditions are controlled by Intel and are not provided in customer documentation.

11.6.6 VLAN Support

Two basic types of VLANs are supported:

- Tagged VLANs are based on the IEEE 802.1Q specification. Each packet has a 4-byte tag added to the packet hearer. The switch must support IEEE 802.1Q tagging and be properly configured. Check your switch documentation for the correct switch configuration.
- Untagged or Port-based VLANs are statically configured on the switch. They are transparent to connected devices.

11.7 Memory-Mapped I/O and Software Interface

The Memory-Mapped I/O (MMIO) register descriptions for each of the four PCI functions of the integrated GbE controller are not in this document. The MMIO register descriptions are in the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*. The PRM also provides details for each function's IOADDR and IODATA registers located in I/O space that can be used to access the controller's internal registers and memory units.

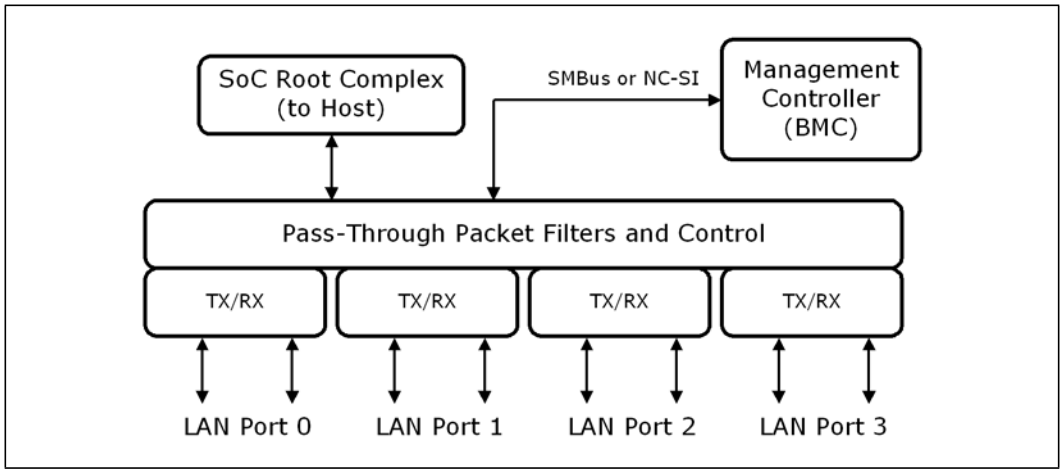
11.8 System Manageability

Network management is an important requirement in today's networked computer environment. Software-based management applications provide the ability to administer systems while the operating system is functioning in a normal power state, that is, when not in a pre-boot state or powered-down state. The System Management Bus (SMBus) Interface, Management Component Transport Protocol (MCTP) and the Network Controller Sideband Interface (NC-SI) fill the management void that exists when the operating system is not running or fully functional. This is accomplished by providing mechanisms by which manageability network traffic can be routed to and from a Management Controller (MC) such as a Baseboard Management Controller (BMC).

The *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)* describes the supported management interfaces and supported hardware configurations for platform system management. It describes the interfaces to an external BMC, the partitioning of platform manageability among system components, and the functionality provided by in each platform configuration. For an overview of the hardware interface, see Section 11.5.13, "SMBus and NC-SI Interface" on page 207.

The SoC integrated GbE controller has the ability to route Ethernet traffic to the host operating system as well as the ability to send Ethernet traffic over the sideband interface to an external BMC. The term "Pass-Through" (PT) is used when referring to the process of the LAN Port sending and receiving Ethernet traffic over the sideband (SMBus or NC-SI) interface to/from the BMC. See Figure 11-3.

Figure 11-3. Manageability Pass-Through



When an Ethernet packet reaches the GbE controller at the LAN Port, it is examined and compared to a number of configurable filters. These filters are configurable by the BMC and include, but not limited to, filtering on:

- MAC Address
- IP Address
- UDP/IP Ports
- VLAN Tags
- EtherType

If the incoming packet matches any of the configured filters, it is passed to the BMC. Otherwise, it is not passed.

Using the MAC Address type of filter, the BMC has at least one dedicated MAC address and incoming Ethernet traffic with the matching MAC address(es) is passed to the BMC. This is the simplest filtering mechanism to use and it allows the BMC to receive all types traffic including, but not limited to, IPMI, NFS, HTTP, etc.

Using the other types of filters, the BMC can share an MAC address (and IP address, if desired) with the Host OS to receive only specific Ethernet traffic. This is useful if the BMC is only interested in specific traffic such as IPMI packets.

Packet-reception flow can also be configured for the packet to be:

- Discarded
- Sent to the Host memory
- Sent to the external BMC
- Sent to both the Host memory and the external BMC

The OS/BMC manageability software model, SMBus commands, NC-SI commands, filtering, and sideband interface control are described in the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*.

11.9 Teaming Support

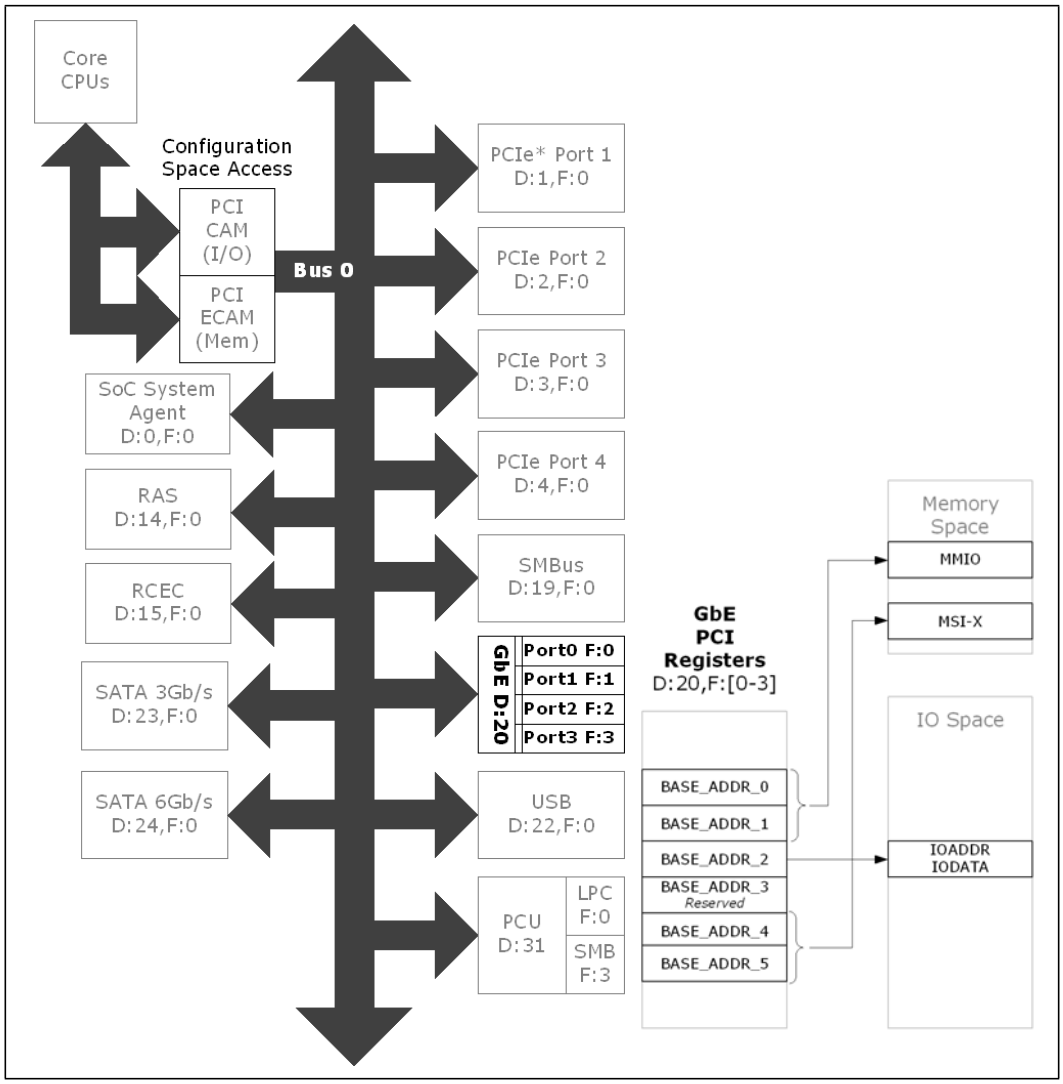
In order to allow the BMC to get all the traffic, it needs to be aware of the teaming event and instruct all ports to receive packets with a destination address matching the MAC addresses of all the channels participating in the team. The BMC is made aware of the teaming event by the reception of a Link Status Change AEN with the Teaming Status bit set. The GbE Interface provides the capability to report Network Interface Controller (NIC) teaming (IEEE 802.3ad) status. Additional information about teaming is in the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*.

11.10 Register Map

The software-accessible registers are outlined in Table 11-3 and are depicted here.

Figure 11-4 shows the SoC GbE Interface registers from a system viewpoint.

Figure 11-4. GbE Interface Register Map



For details of the MMIO and MSI-X registers located in memory space and for the IOADDR and IODATA registers located in IO space, see the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*.

§ §

12 PCI Express Root Ports (RP)

The SoC provides up to four PCI Express* Root Port (RP) Controllers with Gen 2 speeds of up to 5GT/s (per-lane throughput up to 500 MB per second). The interface complies with *PCI Express Base Specification*, Revision 2.1. The root ports are configurable to support a diverse set of lane assignments.The number of available lanes and Root Port controllers depends on the product SKU.

Figure 12-1. PCI Express Root Ports Covered in This Chapter

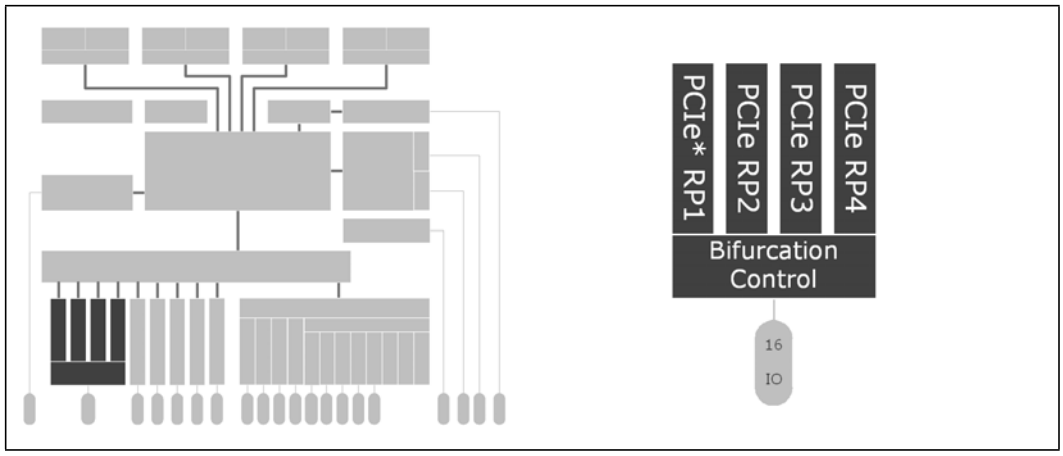


Table 12-1. References

Reference	Revision	Date	Document Title
PCI Express*	2.1	March 4, 2009	*PCI Express Base Specification*, Revision 2.1
PCI-PCI Bridge	1.2	June 9, 2003	*PCI-to-PCI Bridge Architecture Specification*, Revision 1.2
PCI Power Management	1,2	March 3, 2004	*PCI Bus Power Management Interface Specification*, Revision 1.2

12.1 Signal Descriptions

See Chapter 31, "Signal Names and Descriptions," for additional details.

The signal description table has the following headings:

- **Signal Name:** The signal/pin name
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

Note: PMC_WAKE_PCIE# is not listed, but is used by PCI Express* devices. See Chapter 9, "Power Management" for details.

Table 12-2. Signals

Signal Name	Direction/ Type	Description
PCIE_TXP[15:0] PCIE_TXN[15:0]	O PCIe*	PCI Express* Transmit PCI Express Ports 3:0 transmit pair (P and N) signals. Each pair makes up the transmit half of the lane.
PCIE_RXP[15:0] PCIE_RXN[15:0]	I PCIe	PCI Express Receive PCI Express Ports 3:0 receive pair (P and N) signals. Each pair makes up the receive half of the lane.
PCIE_REFCLKP PCIE_REFCLKN	I Differential	PCI Express Input Clock 100 MHz differential clock signal.
PMU_WAKE_B	I PCIe	PCI Express Wake# *This signal is muxed with GPIO_SUS8.*
FLEX_CLK_SE0 FLEX_CLK_SE1	O PCIe	Two, single-ended, flexible, general-purpose clock outputs. Each is default-set to 25 MHz and can be programmed to be 33 MHz or disabled using the CCU Dividers Control Register (DIV_CTRL) located at sideband register Port 40h, offset 0Ch. These signal pins are also used as functional strapping pins during power-up reset. They are used to determine the boot block source (SPI or LPC interface). *These signals are muxed and is used by other functions.*

12.2 Features

The supported features are:

- Inbound INTx interrupts are swizzled
- Supports Reset Warn protocol to ensure a graceful shut down of the Root Port controllers
- Maximum payload size of 256 bytes
- Lane Reversal is inferred during Link training rather than strapped
- Efficient TLP packing with no internal wait states for improved performance
- Store Forward and Threshold Mode are not used
- Request splitting supported on 128/256-byte boundaries
- Routing of Atomic Ops
- System Error Event reporting
- PCIe* Completion Combining (for the same request)
- Interrupts and Events
- Link State support for L0 and L1 (L0s not supported)

The features not supported are:

- PCIe Multicast Capability is not supported.
- Address Translation Services (ATS) are not supported.
- Hardware-based and card-detect PCIe* Hot-Plug* is not supported.
- Vendor-Defined PCIe Messages (VDM) are not supported.
- Relaxed Ordering of a down-stream Completer transaction is not supported.

12.3 Architectural Overview

The SoC supports up to four PCI Express* Gen2 Root Port (RP) controllers. These are discovered by the software in the configuration space on bus 0, devices 1 through 4, each with one function (0). Some product SKUs have only one RP controller.

Each RP controller has three regions in Configuration Space. The contents of these regions and their offset values are:

1. PCI Standard Header
   - Type 1 = PCI-to-PCI bridge
2. PCI Capabilities List
   - PCI Power Management - Capability ID = 01h
   - Message Signaled Interrupts (MSI) - Capability ID = 05h
   - PCI Bridge Subsystem Vendor ID - Capability ID = 0Dh
   - PCI Express - Capability ID = 10h
   - Various implementation-specific and Intel-reserved registers
3. PCI Express Extended Capabilities List
   - Advanced Error Reporting (AER) - Extended Capability ID = 0001h
   - Access Control Services (ACS) - Extended Capability ID = 000Dh
   - Various Vendor-Specific capabilities - Extended Capability ID = 000Bh

The vendor-specific extended capabilities in Configuration Space are primarily for Intel debug and testing purposes.

12.3.1 Atomic Operations (AtomicOps) Routing

This involves a single PCIe agent targeting a location in the memory space and performs a read-modify-write sequence to the location. This technology is supported by the root ports, but not for peer-to-peer transactions.

With PCI Express Atomic Operations (AtomicOps), a PCIe agent performs a single PCI Express transaction targeting a location in memory address space which:

1. Reads the memory location data value.
2. Potentially writes a new value to the data location.
3. Returns the original value.

PCIe Atomic Operations (AtomicOps):

- Fetch and Add (FetchAdd) - The value of a target location is incremented by a specified value using two's complement arithmetic, ignoring any carry or overflow, and the result is written back to the location. The original location value is returned.
- Unconditional Swap (Swap) - A specified value is written to a target location, and the original location value is returned.
- Compare and Swap (CAS) - The value of a target location is compared to a specified value and, if they match, another specified value is written back to the location. Regardless whether they match, the original location value is returned.

Conforming with the *PCI Express Base Specification*, Revision 2.1, the FetchAdd and Swap operations each has one operand. Operand sizes of 32 and 64 bits are supported. The CAS operation has two operands and supports operand sizes of 32, 64, and 128 bits.

Table 12-3. Length Field Values for AtomicOp Requests

AtomicOp	TLP Length Field (in Double-Words) for Particular Operand Size		
	32-Bit Operands	64-Bit Operands	128-Bit Operands
FetchAdd	1	2	N/A
Swap	1	2	N/A
CAS	2	4	8

AtomicOps enable advanced synchronization mechanisms that are particularly useful when multiple producers and/or multiple consumers need to be synchronized in a non-blocking fashion. AtomicOps also enable lock-free statistics counters, for example where a device atomically increments a counter, and host software atomically reads and clears the counter.

Compared to locked transactions:

- AtomicOps provide lower latency, higher scalability, advanced synchronization algorithms, and dramatically less impact to other PCIe traffic.
- Direct support for the three chosen AtomicOps over PCIe, enables easier migration of existing high-performance, Symmetric Multi-Processing (SMP) applications to systems that use PCIe as the interconnect for tightly-coupled co-processors.

Components that implement AtomicOp Requester capability generate any or all three of the AtomicOps, using one or more of the supported operand sizes.

Components that implement AtomicOp Completer capability, carry out the AtomicOps transactions. A PCIe function with AtomicOp Completer capability is not required to support all AtomicOp type/size combinations, and is not required to support all of its Memory Space as a target range for AtomicOps.

For increased interoperability, certain AtomicOp Completer capabilities are required to be supported by root ports in sets if at all.

12.3.2 Reset Warn Technology

Reset Warn is an internal SoC indication to the PCIe Root Port controllers. The SoC ensures a number of conditions are met before applying the reset.

12.3.3 PCI Power Management Capability

The root ports comply with the *PCI Bus Power Management Interface Specification*, Revision 1.2. Refer to the specification for details of the Root Port PCI Power Management Capabilities and register descriptions.

12.3.3.1 Device Power Management States (D-States)

The PCI Express Root Ports support the D0 and D3 states (both $D3_{HOT}$ and $D3_{COLD}$).

The PCI-PM D0, $D3_{HOT}$, $D3_{COLD}$ states correspond to PCI Express link states L0, L1, and L3. When all the PCIe ports and the IOAPIC are programmed to the $D3_{HOT}$ state, the upstream link automatically transitions to the link state L1. The SoC also supports the PME_Turn_Off/PME_TO_Ack PCIe message handshake protocol to enter the $D3_{COLD}$/L3 device/link states.

Each function (downstream ports) behaves as follows when in the $D3_{HOT}$ state:

- The function responds to configuration cycles from upstream.
- The function does not respond to memory cycles from upstream except for Completions.
- The function does not respond to upstream I/O cycles except for completions.The function does not initiate upstream transactions.
- The functions does not reset its registers when programmed from D0 to $D3_{HOT.}$

12.3.3.2 ASPM and ASPM Optionality

The *PCI Express Base Specification*, Revision 2.1 defines the hardware-initiated power management of the PCIe link called the Active State Power Management (ASPM). Under hardware control, the link is in L0 state or an even lower-power L1 link state. ASPM is totally traffic dependent and is not initiated by the software, but the software enables or disables ASPM via the Root Ports PCIe Express Capability structure.

The ASPM Optionality Compliance bit in the Link Capabilities Register of each root port, is used by the software to help determine whether to enable ASPM or whether to run ASPM-compliance tests.

12.3.3.3 Power Management Event (PME) Signaling

At the device level, the SoC root ports support the D0, $D3_{HOT}$ and $D3_{COLD}$ device power management states. In $D3_{HOT}$, the SoC performs a master abort to all configuration requests targeting the functions downstream of the PCI Express Root Port.

- A Power Management Event (PME) is generated from the D0 power state.
- A PME is generated from $D3_{HOT}$ power state.
- A PME is generated from $D3_{COLD}$ power state.
  - The SoC root ports do not support this particular capability, but the capability bit is set for compliance reasons.

12.3.3.4 Beacon and WAKE# Signaling

At the link level, the *PCI Express Base Specification*, Revision 2.1 describes two optional mechanisms used by a components to request the reapplication of main power (transition to the fully-operative L0 state) when in the low-power L2 Link state:

- Beacon (using in-band signaling)
  - Not supported by the SoC root ports.
- WAKE# (using sideband signaling)
  - Supported by the SoC root ports through the SoC input signal pin PMU_WAKE_B.

12.3.3.5 No Soft Reset Bit

When this bit is set, the transition of a PCIe Root Port from $D3_{HOT}$ state to D0 because of a Configuration Write Request to the Power State (PS) field of the Root Port Management Control/Status Register (PMCSR), does not cause an internal soft reset.

12.3.4 PCI Bridge Subsystem Identification Capability

The PCI Bridge Subsystem Vendor ID (Intel® Corporation) and additional identification information are read by the software.

12.3.5 Message Signaled Interrupt (MSI) Capability

Supported by the SoC root ports:

- Per-vector masking capable (PVM).

Not Supported by the SoC root ports:

- Address 64-Bit Capable - The root ports are not capable of generating a 64-bit message address.
- Multiple Message Capable - No, only one message is supported.

Supported by the root ports when enabled by the software [Default]:

- MSI Enable (MSIE) - When set, MSI is enabled and traditional interrupt pins are not used to generate interrupts. Default is 0 (not set).
- Trigger Mode (TM) - Either Edge-Triggered or Level-Triggered. Default is Edge-Triggered.
- Level- or edge-triggered messages are always treated as assert messages. For level-triggered interrupts, the Level bit reflects the interrupt input state if TM (above) specifies the level-triggered mode as follows:
  - 0: Deassert messages
  - 1: Assert messages

12.3.6 Advanced Error Reporting (AER) Capability

PCI Express* defines two error reporting paradigms: the baseline capability and the Advanced Error Reporting (AER) capability. The baseline error reporting capabilities are required of all PCI Express devices and define the minimum error reporting requirements. The SoC root ports provide the optional Advanced Error Reporting Capability which is defined for more robust error reporting and is implemented with a specific PCI Express Capability structure.

PCIe* baseline error handling does not support severity programming. The Advanced Error Reporting Capability provides each of the root ports the Uncorrectable Error Severity register which allows each uncorrectable error to be programmed to fatal or non-fatal. Uncorrectable errors are not recoverable using defined PCI Express mechanisms. However, the SoC considers a particular error fatal to a link or device or possibly considers that error non-fatal. The Uncorrectable Error Severity register (ERRUNCSEV) default value is re-programmed if the device driver or platform software requires more robust error handling.

Note: Peer-to-peer transactions are not supported.

12.3.7 Access Control Services (ACS) Capability

ACS prevents various forms of silent data corruption by preventing PCI Express Requests from being incorrectly routed to a peer Endpoint below a switch. ACS is also used to validate that every request transaction between two downstream components is allowed. Also, ACS allows some robustness checks by checking for the ReqID from a function to be a valid ReqID at a coarse granularity. When ACS is enabled, the PCIe Root Port does loopback memory transactions (reads and writes) to the same port if the address map check matched.

12.4 PCI Configuration Process

After the BIOS initializes the PCI Express Root Ports, during the PCI configuration (Configuration Read and Configuration Write transactions), the root ports respond to Type 0 Configuration Transactions. A Type 0 Configuration Transaction configures the root port (the bridge) and is not forwarded downstream by the bridge (from its primary to secondary interface).

Once the root port is configured, the bridge decides whether to respond to subsequent Type 1 Configuration Transactions. The bridge compares the specified bus number with three configuration registers that were programmed by the configuration initialization code to determine whether to claim and forward a Type 1 configuration transaction across the bridge. For certain bus numbers in a Type 1 Configuration Transaction, the root port converts the transaction to a Type 0 Configuration Transaction and passes the transaction downstream through its secondary interface.

For additional details about PCI bridge operation, see the *PCI-to-PCI Bridge Architecture Specification*, Revision 1.2.

12.4.1 I/O Address Transaction Forwarding

The 8-bit I/O Base field and the 8-bit I/O Limit field in the standard PCI Bridge Header in Configuration Space define how the root port treats PCI I/O-addressed transactions between its primary (Root Complex side) and secondary (downstream link) interfaces.

The I/O Base field and the I/O Limit field are set to indicate that root port only supports 16-bit addressing for I/O-addressed transactions. The root port decodes the full 32 bits of each I/O transaction and only accepts I/O transactions where the address bits [31:16] are zero.

The I/O Base and I/O Limit fields establish the starting and ending I/O addresses the root port accepts for forwarding. Bits [7:4] of these fields define bits [15:12] for I/O transaction addresses for the base and limit addresses. Bits [11:0] for the I/O Base address are fixed as zero. Bits [11:0] for the I/O Limit address are fixed as all ones.

Notice the 4-KB granularity of the forwarding range.

Since the PCIe Root Ports only support 16-bit I/O-address transactions, the I/O Base Upper 16-Bits field and the I/O Base Limit 16-Bits field are not used.

12.4.2 Non-Prefetchable Memory-Address Transaction Forwarding

The 16-bit Memory Base field and the 16-bit Memory Limit field in the standard PCI Bridge Header in Configuration Space define how the root port treats PCI non-prefetchable memory-addressed transactions between its primary (Root Complex side) and secondary (downstream link) interfaces.

Non-prefetchable memory-addressed transactions are typically used for Memory-Mapped I/O (MMIO) access and are always 32-bit-memory-space addresses.

The Memory Base and Memory Limit fields establish the starting and ending memory-mapped addresses the root port accepts for forwarding for non-prefetchable memory transactions. Bits [15:4] of these fields define bits [31:20] for non-prefetchable memory transaction addresses for the Base and Limit addresses. Bits [19:0] for the Memory Base address are fixed as zero. Bits [19:0] for the Memory Limit address are fixed as all ones.

Notice the 1-KB granularity of the forwarding range.

12.4.3 Prefetchable Memory-Address Transaction Forwarding

The following fields in the standard PCI Bridge Header in the configuration space define how the root port treats PCI prefetchable memory-addressed transactions between its primary (Root Complex side) and secondary (downstream link) interfaces.

The root ports support 64-bit prefetchable memory-addressed transactions. The fields are set for this support.

Together with the Prefetchable Base Upper 32-Bits field, the Prefetchable Memory Base field establishes the starting memory-mapped addresses the root port accepts for forwarding prefetchable memory transactions. Likewise, the Prefetchable Limit Upper 32-Bits field and the Prefetchable Memory Limit field establishes the ending address.

As the name implies, the Prefetchable Base Upper 32-Bits field is bits [63:32] of the starting address. Likewise, the Prefetchable Limit Upper 32-Bits field is bits [63:32] of the ending address.

The Prefetchable Memory Base field bits [15:4] define bits [31:20] for the starting address. The Prefetchable Memory Base address bits [19:0] are fixed as zero.

The Prefetchable Memory Limit field bits [15:4] define bits [31:20] for the ending address. The Prefetchable Memory Limit address bits [19:0] are fixed as all ones.

Notice the 1-MB granularity of the forwarding range.

12.4.4 Bus Master Enable (BME) in the Header Command Register

When the Bus Master Enable (BME) bit is set in the Root Port Device PCI Command register (PCICMD) of the PCI Standard Header in Configuration Space, all bridge transactions are treated in a normal fashion. The bridge (the root port) operates as a bus master on the primary interface for memory and I/O transactions forwarded from the secondary interface.

When BME is set to zero, memory read, memory write, I/O read and I/O write requests made in the upstream direction at the root port secondary interface are blocked. Such transactions are handled as Unsupported Requests (UR). For these transactions that are non-posted requests, a completion is also sent when a UR status is returned.

When BME is set to zero, upstream Message Signaled Interrupts (MSI) are also blocked in that they are actually memory write transactions. This also holds true for requests issued by agents using the PCIe message-base mechanism to generate legacy PCI interrupts (INTx).

The BME bit state does not affect the ability of the root port to forward or convert configuration transactions from the secondary interface to the primary interface.

12.5 Interrupts and Events

A root port can handle interrupts and events from an endpoint device. A root port also generates its own interrupts for some events, including power management events, but also including error events.

A root port receives two interrupt types from an endpoint device: INTx (legacy) and MSI. MSIs are automatically passed upstream by the root port, just as other memory writes are passed. INTx messages are delivered to the legacy block interrupt router/controller by the root port.

Events and interrupts that are handled by the root port are shown with the interrupts they deliver to the interrupt decoder/router.

Table 12-4. Interrupts Generated From Events/Packets

Packet/Event	Type	INTx	MSI	SERR	SCI	SMI	GPE
INTx	Packet	X	X				
PM_PME	Packet	X	X				
Power Management (PM)	Event	X	X		X	X	
ERR_CORR	Packet			X			
ERR_NONFATAL	Packet			X			
ERR_FATAL	Packet			X			
Internal Error	Event			X			
VDM	Packet						X

Note: Table 12-4 lists the interrupts and events generated based on packets received, or events generated in the root port. Configuration is needed by the software to enable the different interrupts as applicable.

When INTx interrupts are received by an end point, they are mapped to the interrupts shown in Table 12-5 and sent to the interrupt decoder/router in the Intel SoC legacy block.

Table 12-5. Interrupt Generated for INT[A-D] Interrupts

	INTA	INTB	INTC	INTD
Root Port 1	INTA#	INTB#	INTC#	INTD#
Root Port 2	INTD#	INTA#	INTB#	INTC#
Root Port 3	INTC#	INTD#	INTA#	INTB#
Root Port 4	INTB#	INTC#	INTD#	INTA#

Note: Interrupts generated from events within the root port are not swizzled.

12.5.1 Hot-Plug Events

The SoC PCI Express* Root Ports do not support the hot-add and hot-removal of PCI Express adapters. The PCIe* specification calls this PCI Express Hot-Plug* support. The root ports also do not support the detection of an adapter that has been newly plugged-in (connected) or newly powered-on.

12.5.2 System Error (SERR)

System Error events are supported by both internal and external sources and are mapped to generate either a Non-Maskable Interrupt (NMI) or a System Management Interrupt (SMI). See the *PCI Express Base Specification*, Revision 2.1 for details.

12.6 Power Management

Each root port link supports L0 and L1 link states per *PCI Bus Power Management Interface Specification*, Revision 1.2.

12.7 Physical Layer

12.7.1 PCI Express Speed Support

The PCI Express (PCIe) physical layer implements high-speed, low-voltage differential signaling that is described in detail in the *PCI Express Base Specification*, Revision 2.1. The integrated root ports support 2.5 GT/s and 5 GT/s PCI Express speeds. The Root Port controllers negotiate the speed using the in-band signaling mechanism defined in Section 4.2.4, Link Initialization and Training, of the specification.

The PCI Express Root Ports use 8b/10b encoding when the data rate is 2.5 GT/s or 5 GT/s.

12.7.2 Form Factor Support

The PCIe controllers support card-edge-connector and Server I/O Module (SIOM) form-factors. Form-factor-specific differences that exist for Hot-Plug and power management are captured in their individual sections elsewhere in this chapter.

The root ports have enough buffering to provide full performance using up to a 20-inch trace of FR4 with two connectors. They do not provide any additional buffering for cable/repeater latencies and are not able to achieve full bandwidth on PCI Express using these topologies. But functionally, they are able to support cables. Refer to the *Intel® Atom™ Processor C2000 Product Family - Platform Design Guide (PDG)* for interface topologies supported.

12.8 Configuration of PCI Express Ports and Link Widths

The PCI Express interface consists of up to 16 physical lanes and up to four Root Port controllers, depending on product SKU. For a given SKU, the number of available lanes and controllers may be further reduced and configured using soft straps and bifurcation settings. Table 12-6 shows the supported configurations assuming all 16 lanes and 4 controllers are available.

- Product SKU dictates the maximum number of supported physical lanes and Root Port Controllers. Reference to Section 1.4, "Product SKUs" on page 35.
- Bifurcation may organize the remaining available lanes and controllers.
- During Link training, lane reversal may occur on a per-controller basis.

Table 12-6. Lane and Root Port Controller Configurations

Number of Root Port Controllers Enabled	PCIe* Device Link-Widths Supported (Number of Lanes)	Description
One	x16, x8, x4, x2, x1	For the single controller configuration
Two	x8, x4, x2, x1 and x8, x4, x2, x1	Independent selection per controller
Three	x8, x4, x2, x1 and x4, x2, x1 and x4, x2, x1	Independent selection per controller One wide channel and two narrow channels
All Four	x4, x2, x1 and x4, x2, x1 and x4, x2, x1 and x4, x2, x1	Independent selection per controller

Interfacing with a x8 PCIe device is supported with bifurcation to either:

- One controller as x16.
- Two controllers, each up to x8 wide.
- Three controllers, one up to x8 wide and two up to x4 wide.

With degraded operation:

- A 16-lane link connected to a x8, x2 or x1 PCIe device.
- An 8-lane link connected to a x2 or x1 PCIe device.
- A 4-lane link connected to a x2 or x1 PCIe device.

Since the smallest bifurcation granularity is x4 (for 4 controllers), attached PCIe x2 or x1 devices each require a four-lane controller for connection.

12.8.1 Soft Straps and Bifurcation

Each of the four PCI Express Root Port (RP) controllers is enabled for customer use based on the product SKU tables. A customer-programmable soft strap for each of the enabled RP controllers can be used to further disable RP controller(s). Before the BIOS boot process, the soft strap information (disable) is applied to each RP controller. Refer to *Intel® Atom™ Processor C2000 Product Family SPI Flash Programming Tools and Users Guide* - Doc# 519715.

The 32-bit Root Port Bifurcation Control Register (RP_BIFCTL) in configuration space at bus 0, device 15, function 0, offset 0x40C, provides bifurcation control. The 3-bit Bifurcation Control 0 (BIFCTL0) field of this register can be set to one of five lane arrangements. Bit 16 of this register is the Link Train 0 (LINKTRN0) field which enables link training and bifurcation using values programmed in BIFCTL0. A value of 1 initiates link training. LINKTRN0 must not assert until after BIFCTL0 is programmed.

The control registers are decoded as shown in Table 12-7.

Table 12-7. Bifurcation Control Register

Bifurcation Control 0 (BIFCTL0)	Lane Widths	Lane Numbers Assigned to Each Lane-Width Choice
000	x4 x4 x4 x4	Lanes 15:12 support x4 in RP4 Lanes 11:8 support x4 in RP3 Lanes 7:4 support x4 in RP2 Lanes 3:0 support x4 in RP1
001	x4 x4 x8	Lanes 15:12 support x4 in RP4 Lanes 11:8 support x4 in RP3 (RP2 is unavailable) Lanes 7:0 support x8 in RP1
010	x8 x4 x4	(RP4 is unavailable) Lanes 15:8 support x8 in RP3 Lanes 7:4 support x4 in RP2 Lanes 3:0 support x4 in RP1
011	x8 x8	(RP4 and PR2 are unavailable) Lanes 15:8 support x8 in RP3 Lanes 7:0 support x8 in RP1
100	x16	(RP4, RP3, RP2 are unavailable) Lanes 15:0 support x16 in RP1
101 - 111	Reserved	

A write of 1 to the LINKTRN0 bit locks the setting of the BIFCTL0 register field as the port bifurcation control status information.

After writing this bit to a 1, the software polls the Data Link Layer Link Active (DLLLA) bit in the Link Status Register (LINKSTS) register to determine if a port is up and running. The root ports do not automatically initiate link training after reset unless soft strap default values have already set this bit to 1. A write of 0 has no effect. A write of 1 locks this register bit and initiates link training.

The link widths in degraded mode are shown in Table 12-8.

Table 12-8. Supported Link-Width Matrix in Degraded Mode

Original Link Width[1]	Degraded-Mode Link Width and Lane Numbers
X16	x8 on either lanes 0-7, or 8-15
	x4 on either lanes 0-3, or 12-15
	x2 on either lanes 0-1, or 14-15
	x1 on either lane 0 or 15
X8 Assuming 2 cards present in the system	x4 on either lanes 0-3, 4-7, 8-11 or 12-15
	x2 on either lanes 0-1, 6-7, 8-9, or 14-15
	x1 on either lane 0, 7, 8, or 15
X4 Assuming 4 cards present in the system	x2 on either lanes 0-1, 2-3, 4-5, 6-7, 8-9, 10-11, 12-13, or 14-15
	x1 on either lane 0, 3, 4, 7, 8, 11, 12, or 15

1. This is the native width link that is running at the time the degraded-mode operation kicks-in.

12.8.2 PCI Express Lanes with Various SKUs Design Consideration

In general, most designers want to achieve a single platform design that can be applied to different SoC SKUs. Sometimes it is difficult to implement all feature sets from different SKUs into a single design. Designers need to make a balanced decision about what is important for their design. This section provides design considerations for the PCI Express* configuration support with various SoC SKUs.

12.8.2.1 SoC PCI Express Lanes Mapping

Table 12-9 summarizes the PCIe* lanes and PCIe controller mapping for various SoC SKUs[2].

Table 12-9. PCIe Lanes and PCIe Controller Mapping for Various SKUs

Legend	Description
	PCIe Lane is powered and enabled *
	PCIe Lane is powered but disabled via softstrap Strap 8:SlotWidth Parameter
	PCIe Lane is not available in this SKU
	PCIe Lane is un-powered and disabled via softstraps Strap 0 and Strap 8:PCIe RPx Disable Parameter Strap 5:PCIe Lane y Power Enable Parameter x = port number (1-4), y = lane number (0-15)

PCIe CTRL /Lane Config	Softstrap[1] 0 PCIe RP Disable RP4 RP3 RP2 RP1	Softstrap 8 PCIe RP Disable RP4 RP3 RP2 RP1	Softstrap 5 PCIe Lane Power Enable [15:12][11:8][7:4][3:0]	Softstrap 8 SlotWidth 12 bits total, 3 bits per RP	Bifurcation Controller Register BIFCTL0	Lane 15	Lane 14	Lane 13	Lane 12	Lane 11	Lane 10	Lane 9	Lane 8	Lane 7	Lane 6	Lane 5	Lane 4	Lane 3	Lane 2	Lane 1	Lane 0
x16 Lanes with 4 Controllers SKUs																					
4x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	000	RP4				RP3				RP2				RP1			
1x8, 2x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	001	RP4				RP3				RP1							
2x4, 1x8	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	010	RP3								RP2				RP1			
2x8	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	011	RP3								RP1							
4x16	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	100	RP1															
x8 Lanes with 4 Controllers[3] SKUs																					
4x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	000	RP4				RP3				RP2				RP1			
1x8	4'b1100	4'b1100	16'h0_0_F_F	default = 0_0_0_0	001	RP4				RP3				RP1							
2x4	4'b1100	4'b1100	16'h0_0_F_F	default = 0_0_0_0	010	RP3								RP2				RP1			
1x4	4'b1101	4'b1101	16'h0_0_F_0	default = 0_0_0_0	000	RP4				RP3				RP2				RP1			
4x2	default = 0000	default = 0000	default = FFFF	12'b010_010_010_010	000	RP4				RP3				RP2				RP1			
4x1	default = 0000	default = 0000	default = FFFF	12'b001_001_001_001	000	RP4				RP3				RP2				RP1			
x8 Lanes with 2 Controllers SKUs																					
2x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	000	RP4				RP3				RP2				RP1			
1x8	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	001	RP4				RP3				RP1							
2x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	010	RP3								RP2				RP1			
1x8	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	011	RP3								RP1							
x4 Lanes with 4 Controllers[4] SKUs																					
1x4	4'b1101	4'b1101	16'h0_0_F_0	default = 0_0_0_0	000	RP4				RP3				RP2				RP1			
2x2	4'b1001	4'b1001	16'h0_F_F_0	12'b000_001_001_000	000	RP4				RP3				RP2				RP1			
4x1	default = 0000	default = 0000	default = FFFF	12'b000_001_001_001	000	RP4				RP3				RP2				RP1			
x4 Lanes with 1 Controller SKUs																					
1x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	000	RP4				RP3				RP2				RP1			
1x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	010	RP3								RP2				RP1			

Notes:

1. Soft strap is the SoC strap configuration parameter defined in Flash Descriptor located in the SPI Flash Tool (Flash Image Tool) in the *Intel® Atom™ Processor C2000 Product Family SPI Flash Programming Tools and Users Guide* - document number 519715.
2. See Section 1.4, "Product SKUs" on page 35 to determine if the SKU chosen supports this configuration.
3. This specification requires eight PCIe lanes (out of the total available 16 lanes) to be disabled with soft straps in the SPI Flash.
4. This specification requires twelve PCIe lanes (out of the total available 16 lanes) to be disabled with soft straps in the SPI Flash.

The x16 lanes with four Controllers SKUs, x8 lanes with two Controllers SKUs, and x4 lanes with one Controller SKU, use the Bifurcation Control 0 (BIFCTL0) to configure the PCIe ports and lanes usage with default PCIe soft strap configuration. The x8 lanes with four Controllers SKUs, requires using the PCIe soft strap configuration (soft strap 0, soft strap 5, and soft strap 8) and combine with Bifurcation Control 0 (BIFCTL0) to configure the PCIe ports and lane usage described in Table 12-9.

Table 12-10 shows the supported PCIe lane reversal configurations for Various SKUs[1]. Lane Reversal is determined during Link training.

Table 12-10. Lane Reversal Supported Mapping for Various SKUs

PCIe CTRL /Lane Config	Softstrap[2] 0 PCIe RP Disable RP4 RP3 RP2 RP1	Softstrap 8 PCIe RP Disable RP4 RP3 RP2 RP1	Softstrap 5 PCIe Lane Power Enable [15:12][11:8][7:4][3:0]	Softstrap 8 SlotWidth 12 bits total, 3 bits per RP	Bifurcation Controller Register BIFCTL0	Lane 15	Lane 14	Lane 13	Lane 12	Lane 11	Lane 10	Lane 9	Lane 8	Lane 7	Lane 6	Lane 5	Lane 4	Lane 3	Lane 2	Lane 1	Lane 0
x16 Lanes with 4 Controllers SKUs																					
Base						3			0	3			0	3			0	3			0
4x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	000	RP4				RP3				RP2				RP1			
Lane Reveral						0			3	0			3	0			3	0			3
4x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	000	RP4				RP3				RP2				RP1			
Base						3			0	3			0	7							0
1x8, 2x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	001	RP4				RP3				RP1							
Lane Reveral						0			3	0			3	0							7
1x8, 2x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	001	RP4				RP3				RP1							
Base						7							0	3			0	3			0
2x4, 1x8	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	010	RP3								RP2				RP1			
Lane Reveral						0			7	0			3	0			3	0			3
2x4, 1x8	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	010	RP3								RP2				RP1			
Base						7							0	7							0
2x8	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	011	RP3								RP1							
Lane Reveral						0			7	0			7	0							7
2x8	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	011	RP3								RP1							
Base						16															0
1x16	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	100	RP1															
Lane Reveral						0															16
1x16	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	100	RP1															
x8 Lanes with 4 Controllers[3] SKUs																					
Base								1	0			1	0			1	0			1	0
4x2	default = 0000	default = 0000	default = FFFF	12'b010_010_010_010	000			RP4				RP3				RP2				RP1	
Lane Reveral								0	1			0	1			0	1			0	1
4x2	default = 0000	default = 0000	default = FFFF	12'b010_010_010_010	000			RP4				RP3				RP2				RP1	
x8 Lanes with 2 Controllers SKUs																					
Base						0			3	0			3	7							0
1x8	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	010 or 011	RP4				RP3				RP1							
Lane Reveral						0			3	0			3	0							7
1x8	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	010 or 011	RP4				RP3				RP1							
Base						3			0	3			0	3			0	3			0
2x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	000 or 001	RP4				RP3				RP2				RP1			
Lane Reveral						0			3	0			3	0			3	0			3
2x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	000 or 001	RP4				RP3				RP2				RP1			
x4 Lanes with 1 Controller SKUs																					
Base						3			0	3			0	3			0	3			0
2x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	000 or 010	RP4				RP3				RP2				RP1			
Lane Reveral						0			3	0			3	0			3	3			0
2x4	default = 0000	default = 0000	default = FFFF	default = 0_0_0_0	000 or 010	RP4				RP3				RP2				RP1			

1. See Section 1.4, "Product SKUs" on page 35 to determine if the SKU chosen supports this configuration.
2. Soft strap is the SoC strap configuration parameter defined in the Flash Descriptor located in the SPI Flash Tool *Intel® Atom™ Processor C2000 Product Family SPI Flash Programming Tools and Users Guide* - document number 519715.
3. This specification requires eight PCIe lanes (out of the total available 16 lanes) to be disabled with soft straps in the SPI Flash.

12.9 PCI Express RAS Features

The *PCI Express Base Specification*, Revision 2.1 defines a standard set of error reporting mechanisms. The root ports supports all of them including Error Poisoning and Advanced Error Reporting (AER).

The root ports also support:

- Link-Level Cyclical Redundancy Code (LCRC) and Retry Management as described in Section 3.5. Data Integrity, of the specification.
- Dynamic Link Width (DLW) reduction on link failure. See Section 4.2.4. Link Initialization and Training, of the specification.

12.9.1 Error Detecting, Reporting and Logging

The PCI Express Root Ports provide error detection and logging as specified in the aforementioned specification.

Error messages from the PCI Express Functions or Devices in the hierarchy associated with the root port are detected. The root port, if enabled to do so, reports an interrupt to the CPU. The three classes of errors are:

- Fatal Error (ERR_FATAL Error Message)
  Uncorrectable error conditions which render the particular link and related hardware unreliable. For fatal errors, a reset of the components on the link is required to return to reliable operation.
- Non-Fatal Error (ERR_NONFATAL Error Message)
  Uncorrectable errors which cause a particular transaction to be unreliable but the link is otherwise fully functional. Isolating non-fatal from fatal errors provides requester/receiver logic in a device or system management software the opportunity to recover from the error without resetting the components on the link and disturbing other transactions in progress. Devices not associated with the transaction in error are not impacted by the error.
- Correctable Error (ERR_COR Error Message)
  Correctable errors include those error conditions where the hardware recovers without any loss of information. The hardware corrects these errors and software intervention is not required.

Each error class generates an interrupt to the CPU if the corresponding error class Reporting Enable is set in the Root Port Root Error Command register. This register is located in the configuration space of the root port in its PCI Express Advanced Error Reporting (AER) Extended Capability Structure.

The root port itself as a PCI Express Device also generates the three classes of error messages to the root complex if the particular error class is enabled in the Root Port Device Command register in its PCI Express Capability Registers in Configuration Space.

Another way to enable error reporting by the root port is the Root Port PCI Command register contains SERR# Enable (SEE). When set, this bit enables reporting of non-fatal and fatal errors detected by root port to the root complex. This bit also controls transmission by the root port of ERR_NONFATAL and ERR_FATAL error messages forwarded from the downstream interface. ERR_COR messages are not affected by this bit.

Devices and Functions in the downstream hierarchy that support the AER Capability, and the root ports themselves as devices, also have an Uncorrectable Error Mask register and a Correctable Error Mask register allows each error condition to be masked independently.

12.9.2 Data Poisoning

The PCI Express Root Ports support forwarding poisoned information between the coherent interface and the PCIe link, and vice-versa. The PCIe has a mode where poisoned data is never sent out on PCI Express, i.e., any packet with poisoned data is dropped internally in the root port and an error escalation done.

12.9.3 Link-Level Cyclical Redundancy Code (LCRC)

The PCIe links are 32-bit CRC protected providing for high reliability. The Data Link Layer Packets (DLLPs) utilize a 16-bit CRC scheme. PCIe also provides for a software-transparent recovery from temporary link failures. When received packets are in error, the hardware automatically retransmits the packet.

12.9.4 Link Retraining and Recovery

This also refers to as Dynamic Link Width (DLW) Reduction.

The Root Port PCI Express interface provides a mechanism to recover from a failed link. PCI Express link can operate in a different link width. The SoC supports PCIe port operation in x8, x4, x2, and, in special cases, x1. In case of a persistent link failure, the PCIe link falls back to a smaller link width in attempt to recover from the error. A PCIe x8 link falls back to a x4 link. A PCIe x4 falls back to x2 link, and then to X1 link. This mechanism enables continuation of system operation in case of PCIe link failures.

12.9.5 Unsupported Transactions and Unexpected Completions

If the SoC receives a legitimate PCIe-defined packet that is not included in PCIe supported transactions, then the SoC treats that packet as an unsupported transaction and follows the PCIe rules for handling unsupported requests. If the SoC receives a completion with a requester ID set to the Root Port Requester ID and no matching request is outstanding, then this is considered an unexpected completion.

Also, the SoC detects malformed packets from PCI Express and reports them as errors per the *PCI Express Specification* rules. If the SoC receives a Type 0 Intel-Vendor_Defined message that terminates at the root complex and if the SoC does not recognize this as a valid Intel-supported message, the message is handled by the SoC as an unsupported request with appropriate error escalation (as defined in *PCI Express Specification*). For Type 1 Vendor_Defined messages which terminate at the root complex, the SoC discards the message with no further action.

12.9.6 Unconnected Ports

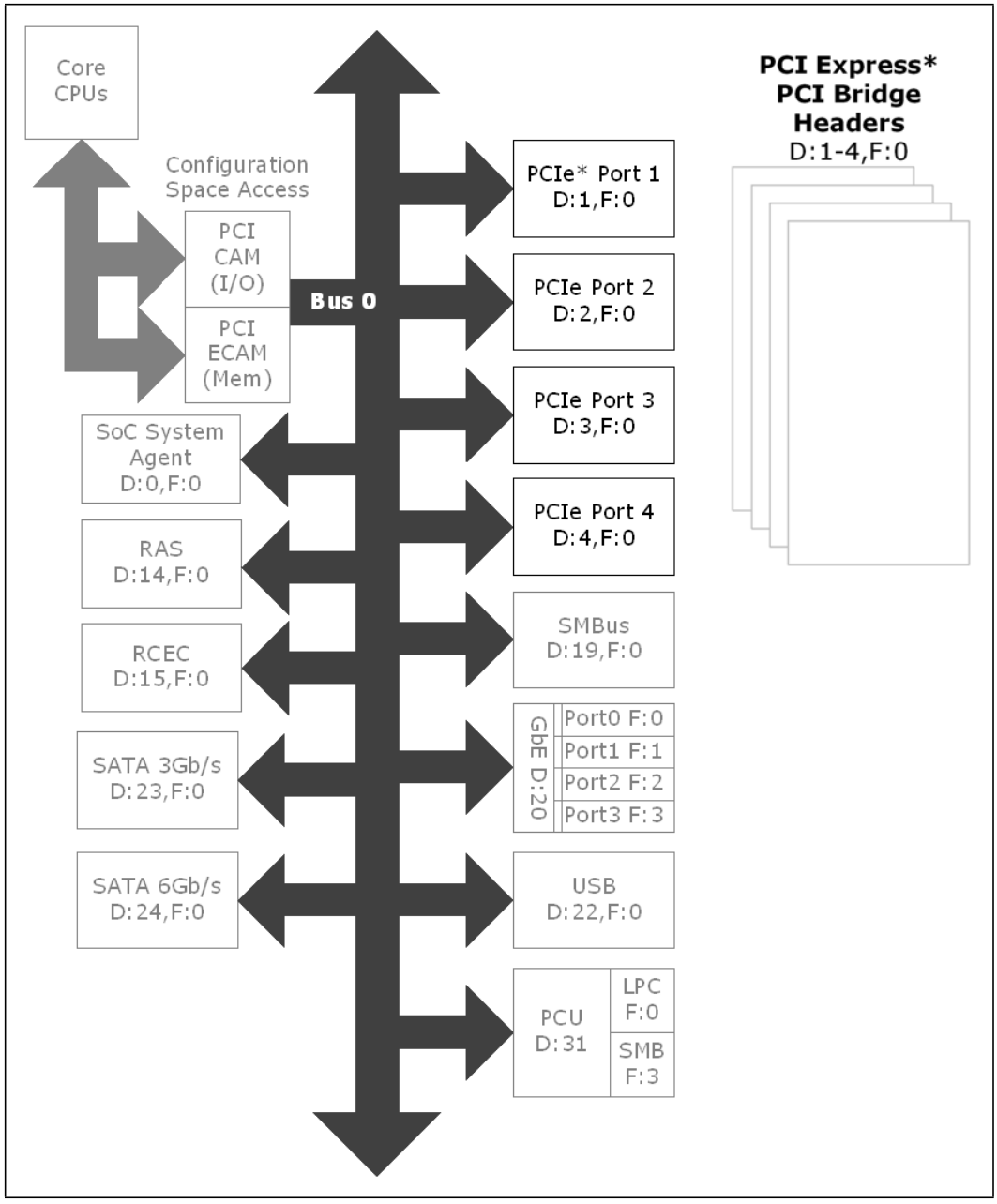
If the local CPU transaction targets a PCIe link that is not connected to any device or the link is down (DL_Down status), the SoC treats that as a master abort situation. This is required for PCI bus scans to non-existent devices to go through without creating any other side effects. If the transaction is non-posted, the SoC synthesizes an unsupported request response status (if non-posted) back to local CPU targeting the unconnected link or returns all Fs on reads and a successful completion on writes targeting the down link.

Note: Accesses by the local CPU to the root port registers.

12.10 Register Maps

Figure 12-2 shows the C2xx0 PCI Express* Root Port registers from a system viewpoint.

Figure 12-2. PCI Express Root Ports Register Map



12.10.1 Registers in Configuration Space

The PCI Express Root Port Controller registers in Configuration Space are shown starting with Table 12-11. A set of these registers exists for each Root Port controller of the SoC. The registers are in the configuration space starting at bus 0, devices 1, 2, 3, and 4, function 0. The offset addresses are listed.

Table 12-11. PCI Standard Type 1 Header

Offset	Name	Description
0x000	VID/DID	DID: 1F10h, 1F11h, 1F12h, 1F13h
0x004	PCICMD	PCI Command Register
0x006	PCISTS	PCI Status Register
0x008	RID	Revision ID Register
0x009	CCR	Class Code Register
0x00C	CLS	Cacheline Size Register
0x00D	PLAT	Primary Latency Timer
0x00E	HDR	Header Type Register
0x00F	BIST	Built-In Self-Test
0x018	PRIBUS	Primary Bus Number Register
0x019	SECBUS	Secondary Bus Number Register
0x01A	SUBBUS	Subordinate Bus Number Register
0x01C	IOBASE	I/O Base Register
0x01D	IOLIMIT	I/O Limit Register
0x01E	SECSTS	Secondary Status Register
0x020	MEMBASE	Memory Base Register
0x022	MEMLIMIT	Memory Limit Register
0x024	PFBASE	Prefetchable Memory Base Register
0x026	PFLIMIT	Prefetchable Memory Limit Register
0x208	PFBASEU	Prefetchable Memory Base Upper 32-Bits Register
0x02C	PFLIMITU	Prefetchable Memory Limit Upper 32-Bits Register
0x034	CAPPTR	Capabilities Pointer Register
0x03C	INTL	Interrupt Line Register
0x03D	INTP	Interrupt Pin Register
0x03E	BCTL	Bridge Control Register

12.10.2 PCI Capabilities

12.10.2.1 PCI Express Capability

Table 12-12. PCI Express Capability

Offset	Name	Description
0x040	EXPCAPLST	PCI Express* Capability List Register
0x042	EXPCAP	PCI Express Capabilities Register
0x044	DEVCAP	Device Capabilities Register
0x048	DEVCTL	Device Control Register
0x04A	DEVSTS	Device Status Register
0x04C	LINKCAP	Link Capabilities Register
0x050	LINKCTL	Link Control Register
0x052	LINKSTS	Link Status Register
0x054	SLOTCAP	Slot Capabilities Register
0x058	SLOTCTL	Slot Control Register
0x05A	SLOTSTS	Slot Status Register
0x05C	ROOTCTL	Root Control Register
0x05E	ROOTCAP	Root Capabilities Register
0x060	ROOTSTS	Root Status Register
0x064	DEVCAP2	Device Capabilities 2 Register
0x068	DEVCTL2	Device Control 2 Register
0x06A	DEVSTS2	Device Status 2 Register
0x06C	LINKCAP2	Link Capabilities 2 Register
0x070	LINKCTL2	Link Control 2 Register
0x072	LINKSTS2	Link Status 2 Register
0x074	SLOTCAP2	Slot Capabilities 2 Register
0x078	SLOTCTL2	Slot Control 2 Register
0x07A	SLOTSTS2	Slot Status 2 Register

12.10.2.2 PCI Power Management Capability

Table 12-13. PCI Power Management Capability

Offset	Name	Description
0x080	PMCAPLST	Power Management Capability List Register
0x082	PMCAP	Power Management Capabilities Register
0x084	PMCSR	Power Management Control/Status Register
0x086	PMBSE	Power Management Bridge Support Extensions Register

12.10.2.3 PCI Bridge Subsystem Vendor ID Capability

Table 12-14. PCI Bridge Subsystem Vendor ID Capability

Offset	Name	Description
0x088	SSCAPLST	Subsystem Capability List Register
0x08C	SSVID	Subsystem Vendor ID Register
0x08E	SSID	Subsystem ID Register

12.10.2.4 Message Signaled Interrupts (MSI) Capability

Table 12-15. Message Signaled Interrupts (MSI) Capability

Offset	Name	Description
0x090	MSICAPLST	MSI Capability List Register
0x092	MSICTL	MSI Message Control Register
0x094	MSIADDR	MSI Message Address Register
0x098	MSIDATA	MSI Message Data Register
0x09C	MSIMSK	MSI Mask Bit Register
0x0A0	MSIPENDING	MSI Pending Bit Register

12.10.3 PCI Express Extended Capabilities

12.10.3.1 Advanced Error Reporting (AER) Extended Capability

Table 12-16. Advanced Error Reporting (AER) Extended Capability

Offset	Name	Description
0x100	AERCAPHDR	Advanced Error Reporting Extended Capability Header
0x104	ERRUNCSTS	Uncorrectable Error Status Register
0x108	ERRUNCMSK	Uncorrectable Error Mask Register
0x10C	ERRUNCSEV	Uncorrectable Error Severity Register
0x110	ERRCORSTS	Correctable Error Status Register
0x114	ERRCORMSK	Correctable Error Mask Register
0x118	AERCAPCTL	Advanced Error Capabilities and Control Register
0x11C - 0x128	AERHDRLOG[1-4]	Header Log Register
0x12C	ROOTERRCMD	Root Error Command Register
0x130	ROOTERRSTS	Root Error Status Register
0x134	ERRSRCID	Error Source Identification Register

12.10.3.2 Access Control Services (ACS) Extended Capability

Table 12-17. Access Control Services (ACS) Extended Capability

Offset	Name	Description
0x138	ACSCAPHDR	Access Control Services Extended Capability Header
0x13C	ACSCAP	Access Control Services Capability Register (ACSCAP)
0x13E	ACSCTL	Access Control Services Control Register
0x140	ERRUNCDETMSK	Uncorrectable Error Detect Mask Register
0x144	ERRCORDETMSK	Correctable Error Detect Mask Register
0x148	ROOTERRDETMSK	Root Error Detect Mask Register

12.10.3.3 Product-Specific Registers

Table 12-18. Product-Specific Registers

Offset	Name	Description
0xEA	PLKCTL	Personality Lock Key Control Register

§ §

13 SATA Controllers (SATA2, SATA3)

The SoC has two independent integrated SATA host controllers. One controller supports DMA operation on up to four ports and supports data transfer rates of 3.0 Gb/s (300 MB/s) and 1.5 Gb/s (150 MB/s). The SATA3 controller in addition to legacy data rates supports data rates of up to 6 Gb/s (600 MB/s). SATA3 controller is the legacy IDE controller and has two ports. Both SATA controllers contains two modes of operation—a native mode and an AHCI mode using memory space. Software that uses a legacy mode does not have AHCI capabilities.

The SoC supports the *Serial ATA Revision 3.0 Specification*. The SoC also supports *Serial ATA Revision 2.6 Specification*.

The SATA controllers feature two sets of interface signals (ports) that are independently enabled or disabled (they cannot be tri-stated or driven low). Each interface is supported by an independent DMA controller.

The SATA controllers interact with an attached mass storage device through a register interface that is equivalent to that presented by a traditional IDE host adapter. The host software follows existing standards and conventions when accessing the register interface and follows standard command protocol conventions.

Note: SATA interface transfer rates are independent of UDMA mode settings. SATA interface transfer rates operate at the bus maximum speed, regardless of the UDMA mode reported by the SATA device or the system BIOS.

Figure 13-1. SATA Controllers Covered in This Chapter

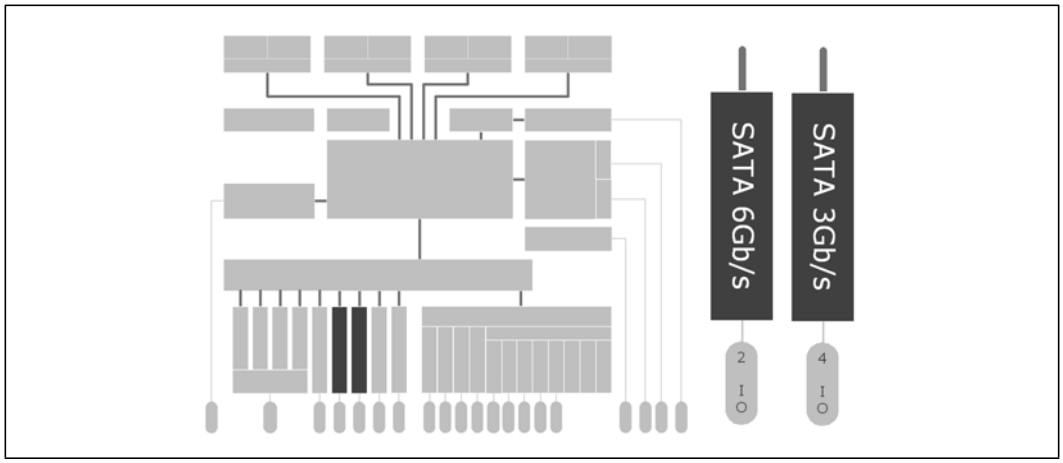


Table 13-1. References

Reference	Revision	Date	Document Title
Serial ATA	3.1	July 18, 2011	*Serial ATA Revision 3.1 Specification*
Serial ATA	3.0	June 2, 2009	*Serial ATA Revision 3.0 Specification*
SATA AHCI	1.3	June 26, 2008	*Serial ATA Advanced Host Controller Interface (AHCI) Revision 1.3 Specification*
Serial ATA	2.6	Feb. 15, 2007	*Serial ATA Revision 2.6 Specification*

13.1 Signal Descriptions

See Chapter 31, "Signal Names and Descriptions" for additional details.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type found in Chapter 31, "Signal Names and Descriptions"
- **Description:** A brief explanation of the signal function

Table 13-2. Signals for SATA2 Interface (Add Signals for SATA3 Interface)

Signal Name	Direction/ Type	Description
SATA_GPI[0]	I/O OD	**Serial ATA 0 General Purpose:** This is an input pin which is configured as an interlock switch or as a general purpose I/O, depending on the platform. When used as an interlock switch status indication, this signal is driven to 0 to indicate that the switch is closed, and to 1 to indicate that the switch is open.
SATA_LEDN	OD	**Serial ATA LED:** This is an open-collector output pin driven during SATA command activity. This pin is to be connected to external circuitry that provides the current to drive a platform LED. When active, the LED is on. When tri-stated, the LED is off.
SATA_TXP[3:0] SATA_TXN[3:0]	O/Differential	**Serial ATA Ports 3:0:** These are outbound high-speed differential signals to Ports 0, 1, 2 and 3.
SATA_RXP[3:0] SATA_RXN[3:0]	I/Differential	**Serial ATA Ports 3:0:** These are inbound high-speed differential signals to Ports 0, 1, 2 and 3.
SATA_REFCLKP SATA_REFCLKN	I/Differential	Serial ATA Controller Clock Input
SATA3_REFCLKP SATA3_REFCLKN	I/Differential	Serial ATA 3 Controller Clock Input

Note: Signals for SATA3 interface are identical to SATA2 interface. The only exception is SATA_TX and SATA_RX signals are 1:0, since only two ports are in the SATA3 controller.

13.2 Features

13.2.1 Supported Features

Table 13-3. SATA Feature List

Feature	Description
Native Command Queuing (NCQ)	Allows the device to reorder commands for more efficient data transfers.
Auto Activate for DMA	Collapses a DMA setup then DMA activate sequence into a DMA setup only.
Asynchronous Signal Recovery	Provides a recovery from a loss of signal or establishing communication after Hot-Plug.
6 Gb/s and 3 Gb/s Transfer Rate	Capable of data transfers up to 6 Gb/s on the SATA3 controller and 3 Gb/s on the SATA2 controller.
ATAPI Asynchronous Notification	A mechanism for a device to send a notification to the host that the device requires attention.
Host and Link Initiated Power Management	Capability for the host controller or device to request partial and slumber interface power states.
Staggered Spin-Up	Enables the host the ability to spin up hard drives sequentially to prevent power load problems on boot.
External SATA	Technology that allows for an outside the box connection of up to 2 meters (when using the cable in SATA-I/O). ***Note:*** This feature in NOT supported for SATA3 controller ports.
SATA2 IDE Legacy Mode	Not supported within the SATA2 controller.
SATA3 IDE Legacy Mode	Supported only within the SATA3 controller.

Table 13-4. SATA/AHCI Feature Matrix

Feature	AHCI Disabled	AHCI Enabled
Native Command Queuing (NCQ)	N/A	Supported
Auto Activate for DMA	N/A	Supported
Asynchronous Signal Recovery	N/A	Supported
3 Gb/s Transfer Rate	Supported	Supported
ATAPI Asynchronous Notification	N/A	Supported
Host and Link Initiated Power Management	N/A	Supported
Staggered Spin-Up	Supported	Supported
6 Gb/s Transfer Rate on the SATA3 Controller	Supported	Supported
External SATA	N/A	Supported

13.2.2 Theory of Operation

13.2.2.1 Standard ATA Emulation

The SoC contains a set of registers that shadow the contents of the legacy IDE registers. The behavior of the Command and Control Block registers, PIO, and DMA data transfers, resets, and interrupts are all emulated.

Note: The SoC asserts INTR when the master device completes the EDD command regardless of the command completion status of the slave device. If the master completes EDD first, an INTR is generated and BSY remains 1 until the slave completes the command. If the slave completes EDD first, BSY is 0 when the master completes the EDD command and asserts INTR. The software must wait for busy to clear (0) before completing an EDD command, as required by the ATA5 through ATA7 (T13) industry standards.

13.2.2.2 48-Bit LBA Operation

The SATA host controller supports 48-bit LBA through the host-to-device register FIS when accesses are performed using writes to the task file. The SATA host controller ensures that the correct data is put into the correct byte of the host-to-device FIS.

Special considerations exist when reading from the task file to support 48-bit LBA operation. The software may need to read all 16 bits. Since the registers are only 8-bits wide and act as a FIFO, a bit must be set in the device/control register, which is at offset 3F6h for primary and 376h for secondary (or their native counterparts).

If the software clears bit 7 of the control register before performing a read, the last item written is returned from the FIFO. If the software sets bit 7 of the control register before performing a read, the first item written is returned from the FIFO.

13.2.3 SATA Swap Bay Support

The SoC provides for basic SATA swap bay support using the PSC register configuration bits and power management flows. A device is powered down by the software and the port is then disabled, allowing removal and insertion of a new device.

Note: This SATA swap bay operation requires board hardware (implementation specific), BIOS, and operating system support.

13.2.4 Function Level Reset Support (FLR)

The SATA host controller supports the Function Level Reset (FLR) capability. The FLR capability is used in conjunction with Intel® Virtualization Technology. FLR allows an operating system in a virtual machine to have complete control over a device, including its initialization, without interfering with the rest of the platform. The device provides a software interface that enables the operating system to reset the whole device as if a PCI reset was asserted.

13.2.4.1 FLR Steps

13.2.4.1.1 FLR Initialization

1. A FLR is initiated by the software writing a 1 to the Initiate FLR bit.
2. All subsequent requests targeting the function are not claimed and are master abort immediate on the bus. This includes any configuration, I/O, or memory cycles; however, the function continues to accept completions targeting the function.

13.2.4.1.2 FLR Operation

The function resets all configuration, I/O and memory registers of the function except those indicated otherwise and reset all internal states of the function to the default or initial condition.

13.2.4.1.3 FLR Completion

The Initiate FLR bit is reset (cleared) when the FLR reset is completed. This bit indicates to the software that the FLR reset is completed.

13.2.5 Power Management Operation

Power management of the SATA controller and ports covers operations of the host controller and the SATA wire.

13.2.5.1 Power State Mappings

The D0 PCI power management state for device is supported by the SATA controller.

SATA devices also have multiple power states. From parallel ATA, three device states are supported through ACPI. They are:

- **D0** – Device is working and instantly available.
- **D1** – Device enters when receiving a STANDBY IMMEDIATE command. Exit latency from this state is in seconds.
- **D3** – From the SATA device perspective, no different than a D1 state, in that it is entered using the STANDBY IMMEDIATE command. However, an ACPI method is also called which resets the device and then cut its power.

These device states are subsets of the host controller D0 state.

Finally, SATA defines three PHY layer power states, which have no equivalent mappings to parallel ATA. They are:

- **PHY READY** – PHY logic and PLL are both on and active.
- **Partial** – PHY logic is powered, but in a reduced state. Exit latency is no longer than 10 ns.
- **Slumber** – PHY logic is powered, but in a reduced state. Exit latency is up to 10 ms.

Since these states have much lower exit latency than the ACPI D1 and D3 states, the SATA controller defines these states as sub-states of the device D0 state.

13.2.5.2 Power State Transitions

13.2.5.2.1 Partial and Slumber State Entry/Exit

The partial and slumber states save interface power when the interface is idle. It is most analogous to PCI CLKRUN# (in power savings, not in mechanism), where the interface has power saved while no commands are pending. The SATA controller defines PHY layer-power management (as performed using primitives) as a driver operation from the host side, and a device proprietary mechanism on the device side. The SATA controller accepts device transition types, but does not issue any transitions as a host. All received requests from a SATA device are ACKed.

When an operation is performed to the SATA controller and needs to use the SATA cable, the controller must check whether the link is in the partial or slumber states, and if so, must issue a COM_WAKE to bring the link back online. Similarly, the SATA device must perform the same action.

13.2.5.2.2 Device D1, D3 States

These states are entered after some period of time when the software has determined that no commands are sent to this device for some time. The mechanism for putting a device in these states does not involve any work on the host controller, other then sending commands over the interface to the device. The command most likely to be used in ATA/ATAPI is the STANDBY IMMEDIATE command.

13.2.5.2.3 Host Controller $D3_{HOT}$ State

After the interface and device have been put into a low-power state, the SATA host controller is put into a low-power state. This is performed using the PCI power management registers in configuration space.

Note: Two important aspects when using PCI power management:

1. When the power state is D3, only accesses to configuration space are allowed. Any attempt to access the memory or I/O spaces results in master abort.
2. When the power state is D3, no interrupts are generated, even if they are enabled. If an interrupt status bit is pending when the controller transitions to D0, an interrupt is generated.

When the controller is put into D3, the assumption is that the software has properly shut down the device and disabled the ports. Sustaining any values on the port wires is not needed. The interface is treated as if a device is not present on the cable, and power is minimized.

When returning from a D3 state, an internal reset is not performed.

13.2.5.2.4 Non-AHCI Mode PME# Generation

When in non-AHCI mode (legacy mode) of operation, the SATA controller does not generate PME#. This includes attach events (since the port must be disabled), or interlock switch events (using the SATA_GP0 pin).

13.2.5.3 SMI Trapping (APM)

The ATC register in configuration space contains control for generating SMI# on accesses to the IDE I/O spaces. These bits map to the legacy ranges (1f0h-1f7h, 3f4h-3f6h, 170h-177h, and 374h-376h) and native IDE ranges defined by PCMDBA, PCTLBA, SCMDBA and SCTLBA. Trapping does not occur on the native Bus Master IDE ranges defined by LBAR. If the SATA controller is in legacy mode and is using these addresses, accesses to one of these ranges with the appropriate bit set cause the cycle to not be forwarded to the SATA controller, and for SMI# to be generated.

Additionally, an ATS register bit is set on an access to these ranges irrespective of the corresponding bit in the ATC register. SMI generation is dependent the corresponding bit in the ATC register being set.

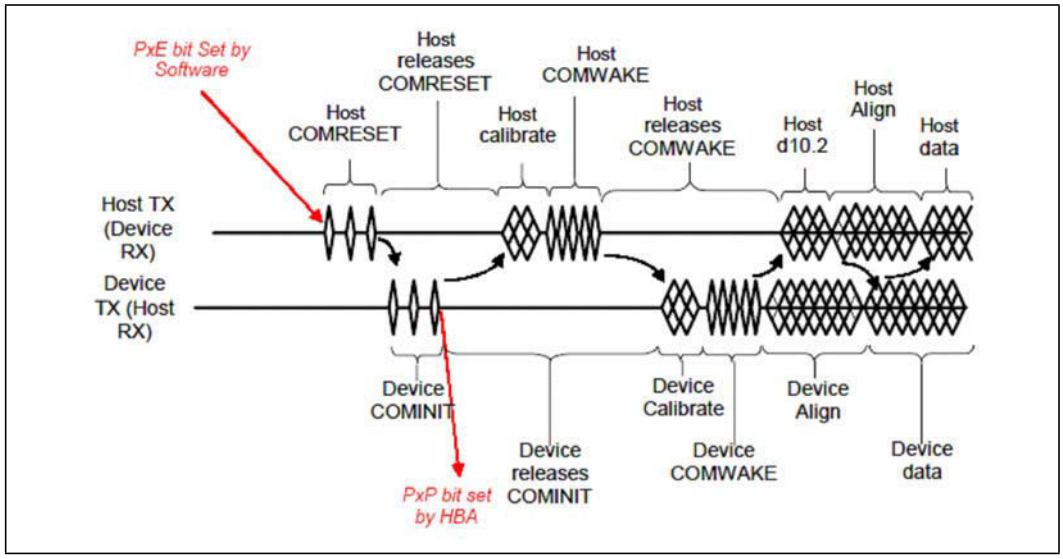
Note: To the BIOS: the Global SMI Enable (SMI_EN.GBL_SMI_EN) register bit in the Power Management register space must be 1 when ATC=1. If the BIOS intends to clear SMI_EN.GBL_SMI_EN to 0, the BIOS clears ATC to 0 as well to avoid a hang condition when the trap range is accessed.

13.2.6 SATA Device Presence

The SATA PHY does know when a device is connected (if not in a partial or slumber state), and its beneficial to communicate this information to host software as this greatly reduces boot times and resume times.

The flow used to indicate SATA device presence is shown in Figure 13-2. The PxE bit refers to PCS.P[1:0]E bits, depending on the port being checked and the PxP bits refer to the PCS.P[1:0]P bits, depending on the port being checked. If the PCS/PxP bit is set a device is present, if the bit is cleared a device is not present. If a port is disabled, the software checks if a new device is connected by periodically re-enabling the port and observing if a device is present. If a device is not present, the software disables the port and checks again later. If a port remains enabled, the software periodically polls PCS.PxP to verify if a new device is connected.

Figure 13-2. Flow for Port Enable/Device Present Bits



13.2.7 SATA LED

The SATA_LED# output is driven whenever the BSY bit is set in any SATA port. The SATA_LED# is an active-low, open-drain output. When SATA_LED# is low, the LED is active. When SATA_LED# is high, the LED is inactive.

13.2.8 AHCI Operation

The SoC provides hardware support for the Advanced Host Controller Interface (AHCI), a programming interface for SATA host controllers developed through a joint industry effort. AHCI defines transactions between the SATA controller and the software and enables advanced performance and usability with SATA. Platforms supporting AHCI take advantage of performance features such as no master/slave designation for SATA devices—each device is treated as a master—and hardware-assisted native command queuing. AHCI requires the appropriate software support (such as, an AHCI driver) and for some features, hardware support in the SATA device or additional platform hardware.

The SoC supports all of the mandatory features of the *Serial ATA Advanced Host Controller Interface (AHCI) Revision 1.3 Specification* and many optional features, such as hardware-assisted native command queuing, aggressive power management, LED indicator support, and Hot-Plug through the use of interlock switch support (additional platform hardware and software are required depending upon the implementation).

Note: For reliable device removal notification while in AHCI operation without the use of interlock switches (surprise removal), interface power management is disabled for the associated port. See Section 7.3.1 of the *Serial ATA Advanced Host Controller Interface (AHCI) Revision 1.3 Specification* for more information.

13.2.9 External SATA

The SATA2 controller in SoC supports external SATA. External SATA utilizes the SATA interface outside of the system box. The usage model for this feature must comply with the *Serial ATA II Cables and Connectors Volume 2 Gold Specification* at www.sata-io.org. Intel validates two configurations:

1. The cable-up solution involves an internal SATA cable that connects to the SATA motherboard connector and spans to a back panel PCI bracket with an eSATA connector. A separate eSATA cable is required to connect an eSATA device.
2. The back-panel solution involves running a trace to the I/O back panel and connecting a device using an external SATA connector on the board.

13.3 Staggered Spin-Up Support

The staggered spin-up feature enables a controller to individually spin-up attached devices. This mechanism is useful to avoid having a power supply that must handle a maximum current draw from all devices and at the same time applying power to the devices. In order for a system to support staggered spin-up, the SATA controller, the BIOS, and the device driver must all support staggered spin-up.

Staggered spin-up supported in both legacy and AHCI modes, is controlled by the software via sequencing of COMRESET signaling. Table 13-5 summarizes the operations of SATA controller in legacy and AHCI modes. Refer to the *AHCI Specification* for more details on AHCI staggered spin-up.

Table 13-5. Operations Summary of SATA Controller in Legacy and AHCI Modes

MAP.SMS	PCS.PxE	GHC.AE	CAP.SSS	PxCMD.SUD	Result
00b	0	x	x	x	No COMRESET and no spin-up.
00b	1	x	x	x	COMRESET and spin-up.
01b/10b	0	x	x	x	No COMRESET and no spin-up.
01b/10b	1	1	0	0	This is not a valid combination. When CAP.SSS is 0, PxCMD.SUD must be 1.
01b/10b	1	1	0	1	COMRESET and spin-up.
01b/10b	1	1	1	0	No COMRESET and no spin-up.
01b/10b	1	1	1	1	COMRESET and spin-up.

13.3.1 Staggered Spin-Up Operations in IDE Mode

When configured the controller in IDE mode (CC.SCC = 01h), the device driver or BIOS has to cycle through PCS.PxE bits. COMRESET is sent for the ports whose PCS.PxE bit is 1. To ensure that the target device is ready and completely spun, the BIOS/device driver is to check if BSY is cleared. The device driver/BIOS is to check for the PCS.PxP bit (set to 1) to determine if the device is present. If the PCS.PxP bit is not set with 10 ms of setting the PCS.PxE bit, this indicates that no target device is connected to the port.

13.3.2 Staggered Spin-Up Operation in AHCI Mode

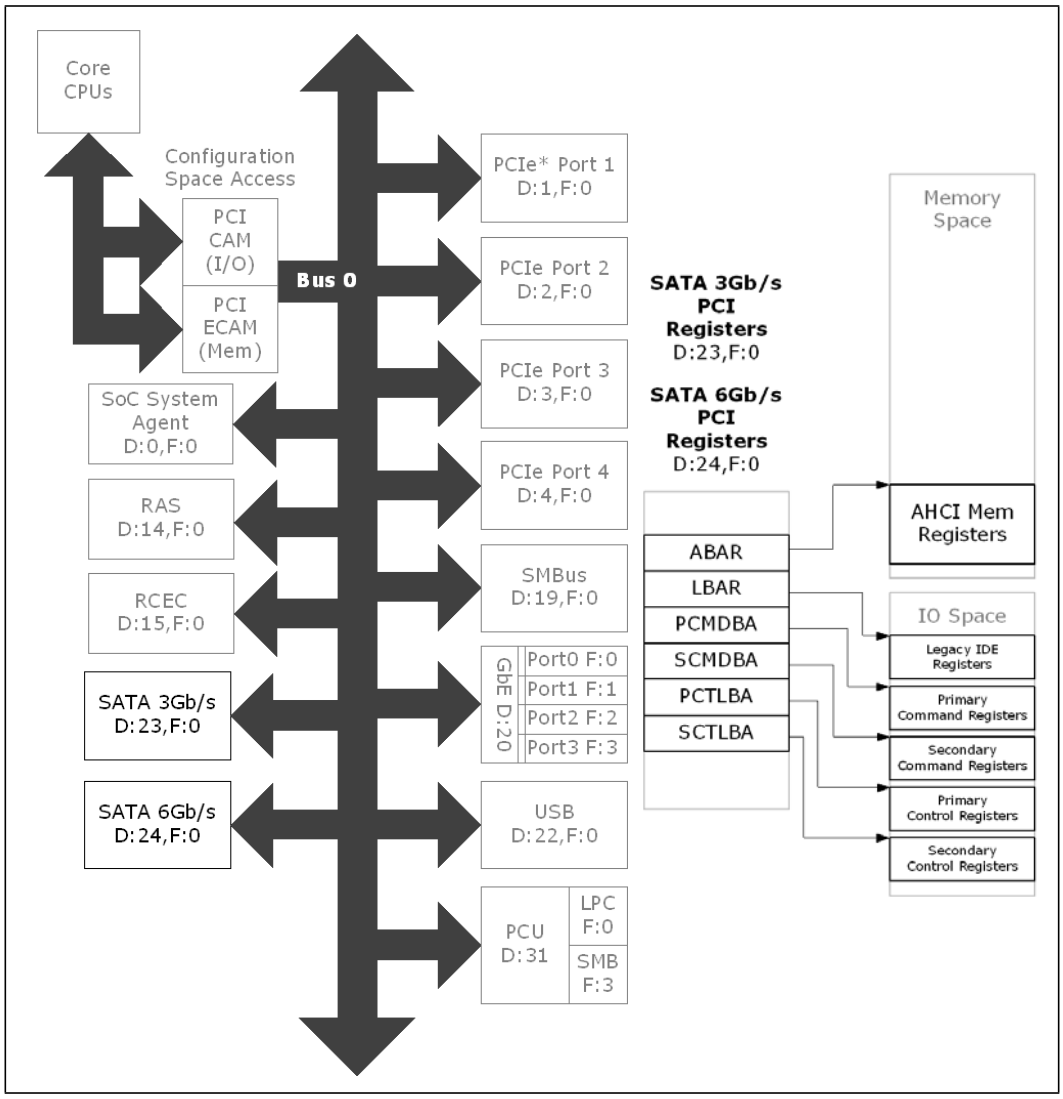
The device driver/BIOS is to perform the following actions:

1. Program ABAR.
2. Configure the controller in the AHCI mode by setting the AHCI GHC.AE bit.
3. Set AHCI CAP.SSS to indicate a staggered spin-up support.
4. Set the PCS.PxE bits.
5. Set the PxCMD.SUD bits as this results in COMRESET to be issued to all the enabled ports.
6. Poll PxSSTS.DET to obtain the status of the port.

13.4 Register Map

Figure 13-3 shows the SoC SATA controller registers from a system viewpoint.

Figure 13-3. SATA Register Map



13.5 PCI Configuration Registers

All of the SATA configuration registers are in the core well. All registers not mentioned are reserved.

All configuration registers are reset by Function Level Reset (FLR) unless specified otherwise explicitly.

Table 13-6. Summary of PCI Configuration Registers—0x_00_13_00 (Sheet 1 of 2)

Offset Start	Offset End	Register ID—Description
0h	3h	"ID (ID)—Offset 0h"
4h	5h	"CMD (CMD)—Offset 4h"
6h	7h	"STS (STS)—Offset 6h"
8h	8h	"RID (RID)—Offset 8h"
9h	9h	"PI (PI)—Offset 9h"
Ah	Bh	"CC (CC)—Offset Ah"
Ch	Ch	"CLS (CLS)—Offset Ch"
Dh	Dh	"MLT (MLT)—Offset Dh"
Eh	Eh	"HTYPE (HTYPE)—Offset Eh"
10h	13h	"PCMDBA (PCMDBA)—Offset 10h"
14h	17h	"PCTLBA (PCTLBA)—Offset 14h"
18h	1Bh	"SCMDBA (SCMDBA)—Offset 18h"
1Ch	1Fh	"SCTLBA (SCTLBA)—Offset 1Ch"
20h	23h	"LBAR (LBAR)—Offset 20h"
24h	27h	"ABAR (ABAR)—Offset 24h"
2Ch	2Fh	"SS (SS)—Offset 2Ch"
34h	34h	"CAP (CAP)—Offset 34h"
3Ch	3Fh	"INTR (INTR)—Offset 3Ch"
40h	41h	"PTIM (PTIM)—Offset 40h"
42h	43h	"STIM (STIM)—Offset 42h"
44h	44h	"D1TIM (D1TIM)—Offset 44h"
48h	48h	"Synchronous_DMA_Control (Synchronous_DMA_Control)—Offset 48h"
4Ah	4Bh	"Synchronous_DMA_Timing (Synchronous_DMA_Timing)—Offset 4Ah"
54h	57h	"IIOC (IIOC)—Offset 54h"
70h	71h	"PID (PID)—Offset 70h"
72h	73h	"PC (PC)—Offset 72h"
74h	75h	"PMCS (PMCS)—Offset 74h"
80h	81h	"MID (MID)—Offset 80h"
82h	83h	"MC (MC)—Offset 82h"
84h	87h	"MA (MA)—Offset 84h"
88h	89h	"MD (MD)—Offset 88h"
90h	91h	"MAP (MAP)—Offset 90h"
92h	93h	"PCS (PCS)—Offset 92h"
94h	97h	"TM (TM)—Offset 94h"
98h	9Bh	"TM2 (TM2)—Offset 98h"

Table 13-6. Summary of PCI Configuration Registers—0x_00_13_00 (Sheet 2 of 2)

Offset Start	Offset End	Register ID—Description
9Ch	9Fh	"SATAGC (SATAGC)—Offset 9Ch"
A0h	A0h	"SIRI (SIRI)—Offset A0h"
A4h	A7h	"SIRD (SIRD)—Offset A4h"
A8h	ABh	"SATACR0 (SATACR0)—Offset A8h"
ACh	AFh	"SATACR1 (SATACR1)—Offset ACh"
B0h	B1h	"FLR Capability ID (FLRCID)—Offset B0h"
B2h	B3h	"FLR Capability Length and Version (FLRCAP)—Offset B2h"
B4h	B5h	"FLR Control (FLRCTL)—Offset B4h"
C0h	C0h	"ATC (ATC)—Offset C0h"
C4h	C4h	"ATS (ATS)—Offset C4h"
D0h	D3h	"SP (SP)—Offset D0h"
E0h	E3h	"BFCS (BFCS)—Offset E0h"
E4h	E7h	"BFTD1 (BFTD1)—Offset E4h"
E8h	EBh	"BFTD2 (BFTD2)—Offset E8h"
F8h	FBh	"MFID (MFID)—Offset F8h"
FCh	FFh	"PCMDIDEBA (PCMDIDEBA)—Offset FCh"
100h	103h	"SCMDIDEBA (SCMDIDEBA)—Offset 100h"
104h	107h	"PCTLIDEBA (PCTLIDEBA)—Offset 104h"
108h	10Bh	"SCTLIDEBA (SCTLIDEBA)—Offset 108h"

13.6 Bus Master IDE I/O Registers

This controller implements IDE Bus Master registers and ATA task file shadow registers. All these I/O registers are in the core well. All I/O registers are reset by FLR.

Table 13-7 contains the IDE Bus-Master registers which are accessible via LBAR. Registers with offsets of 00h to 0Ch are commonly termed as IDE bus master registers that occupy 16 bytes of I/O space and are only used for legacy operation. When CC.SCC is 01h, only the bus master registers are mapped; the INDEX and DATA registers are not mapped and not accessible; only 16 bytes of I/O space are allocated. When CC.SCC is not 01h, all registers are mapped and accessible; 32 bytes of I/O space are allocated.

Table 13-7. Summary of I/O Registers—LBAR

Offset Start	Offset End	Register ID—Description
0h	0h	"PCMD (PCMD)—Offset 0h"
2h	2h	"PSTS (PSTS)—Offset 2h"
4h	7h	"PDTP (PDTP)—Offset 4h"
8h	8h	"SCMD (SCMD)—Offset 8h"
Ah	Ah	"SSTS (SSTS)—Offset Ah"
Ch	Fh	"SDTP (SDTP)—Offset Ch"
10h	13h	"INDEX (INDEX)—Offset 10h"
14h	17h	"DATA (DATA)—Offset 14h"

13.7 Serial ATA Index/Data Pair Superset Registers

All of these I/O registers are in the core well. They are exposed only when CC.SCC is 01h (i.e., IDE programming interface).

These are Index/Data Pair registers that are used to access the SerialATA superset registers (SerialATA Status, SerialATA Control and SerialATA Error). The I/O space for these registers is allocated through SIDPBA. Locations with offset from 08h to 0Fh are reserved for future expansion. Software write operations to the reserved locations has no effect while the software read operations to the reserved locations return 0.

Table 13-8. Summary of I/O Registers—ABAR

Offset Start	Offset End	Register ID—Description
0h	3h	"SINDX (SINDX)—Offset 0h"
4h	7h	"SDATA (SDATA)—Offset 4h"

13.8 Memory-Mapped Registers

All of the AHCI memory-mapped registers are in the core well unless stated otherwise. The memory-mapped registers within the SATA controller exist in non-cacheable memory space. Additionally, locked accesses are not supported. If the software attempts to perform locked transactions to the registers, indeterminate results occur. Register accesses have a maximum size of 64 bits. The 64-bit accesses must not cross an 8-byte alignment boundary.

The registers are divided into two sections – global control registers and port control registers. All registers that start below address 100h are global and meant to apply to the entire HBA. The port control registers are the same for all ports, and there are as many registers banks as there are ports.

All registers not defined and all reserved bits within registers return 0 when read. These registers are not accessible when CC.SCC is 01h.

All memory registers are reset by FLR unless specified otherwise.

Note: The memory map registers below are for SATA port 0 and 1 for both SATA2 and SATA3 controllers. Memory map register information for ports 2 and 3 for the SATA2 controller are identical to ports 0 and 1. Register offset information is added in the later revisions.

Table 13-9. Summary of Memory-Mapped I/O Registers—ABAR (Sheet 1 of 2)

Offset Start	Offset End	Register ID—Description
0h	3h	"GHC_CAP (GHC_CAP)—Offset 0h"
4h	7h	"GHC (GHC)—Offset 4h"
8h	Bh	"IS (IS)—Offset 8h"
Ch	Fh	"GHC_PI (GHC_PI)—Offset Ch"
10h	13h	"VS (VS)—Offset 10h"
1Ch	1Fh	"EM_LOC (EM_LOC)—Offset 1Ch"
20h	23h	"EM_CTL (EM_CTL)—Offset 20h"
24h	27h	"GHC_CAP2 (GHC_CAP2)—Offset 24h"
A0h	A3h	"VSP (VSP)—Offset A0h"
A4h	A7h	"VS_CAP (VS_CAP)—Offset A4h"
C4h	C5h	"PFB (PFB)—Offset C4h"
C8h	C9h	"SFM (SFM)—Offset C8h"
100h	103h	"PxCLB (PxCLB0)—Offset 100h"
104h	107h	"PxCLBU (PxCLBU0)—Offset 104h"
108h	10Bh	"PxFB (PxFB0)—Offset 108h"
10Ch	10Fh	"PxFBU (PxFBU0)—Offset 10Ch"
110h	113h	"PxIS (PxIS0)—Offset 110h"
114h	117h	"PxIE (PxIE0)—Offset 114h"
118h	11Bh	"PxCMD (PxCMD0)—Offset 118h"
120h	123h	"PxTFD (PxTFD0)—Offset 120h"
124h	127h	"PxSIG (PxSIG0)—Offset 124h"
128h	12Bh	"PxSSTS (PxSSTS0)—Offset 128h"
12Ch	12Fh	"PxSCTL (PxSCTL0)—Offset 12Ch"
130h	133h	"PxSERR (PxSERR0)—Offset 130h"

Table 13-9. Summary of Memory-Mapped I/O Registers—ABAR (Sheet 2 of 2)

Offset Start	Offset End	Register ID—Description
134h	137h	"PxSACT (PxSACT0)—Offset 134h"
138h	13Bh	"PxCI (PxCI0)—Offset 138h"
144h	148h	"PxDEVSLP (PxDEVSLP0)—Offset 144h"
180h	183h	"PxDEVSLP (PxDEVSLP0)—Offset 144h"
184h	187h	"PxCLBU (PxCLBU1)—Offset 184h"
188h	18Bh	"PxFB (PxFB1)—Offset 188h"
18Ch	18Fh	"PxFBU (PxFBU1)—Offset 18Ch"
190h	193h	"PxIS (PxIS1)—Offset 190h"
194h	197h	"PxIE (PxIE1)—Offset 194h"
198h	19Bh	"PxCMD (PxCMD1)—Offset 198h"
1A0h	1A3h	"PxTFD (PxTFD1)—Offset 1A0h"
1A4h	1A7h	"PxSIG (PxSIG1)—Offset 1A4h"
1A8h	1ABh	"PxSSTS (PxSSTS1)—Offset 1A8h"
1ACh	1AFh	"PxSCTL (PxSCTL1)—Offset 1ACh"
1B0h	1B3h	"PxSERR (PxSERR1)—Offset 1B0h"
1B4h	1B7h	"PxSACT (PxSACT1)—Offset 1B4h"
1B8h	1BBh	"PxCI (PxCI1)—Offset 1B8h"
1C4h	1C7h	"PxDEVSLP (PxDEVSLP1)—Offset 1C4h"
580h	583h	"EM_MF (EM_MF)—Offset 580h"
584h	587h	"EM_LED (EM_LED)—Offset 584h"

§ §

14 Universal Serial Bus (USB) 2.0

The SoC contains one Enhanced Host Controller Interface (EHCI) and complies to the *EHCI 1.0 Specification*. The EHCI supports up to four USB 2.0 root ports. USB 2.0 allows data transfers up to 480 Mbps. The controller integrates a Rate-Matching Hub (RMH) to support USB 1.1 devices. The USB Port 1 interface is configured by the debug software to be a debug port.

In this document, the USB 2.0 EHCI-compliant host controller is referred to as the Enhanced Host Controller (EHC).

Figure 14-1. USB Covered in This Chapter

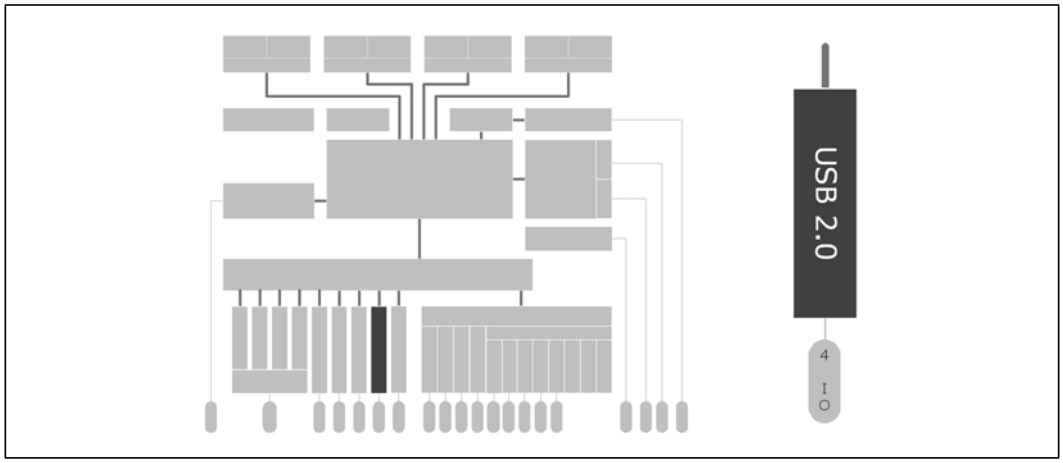


Table 14-1. References

Reference	Revision	Date	Document Title
USB 2.0	2.0	April 27, 2000	*Universal Serial Bus Specification*, Revision 2.0
Intel EHCI	1.0	March 12, 2002	*Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0

14.1 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type found in Chapter 31, "Signal Names and Descriptions"
- **Description:** A brief explanation of the signal function

Table 14-2. Signals

Signal Name	Direction / Type	Description
USB_DP[3:0] USB_DN[3:0]	I/O	**Universal Serial Bus Port 3:0 Differentials:** Bus Data/Address Command Bus
USB_OC0_B	I	**Over-Current Indicator:** This signal sets the corresponding bit in the USB controller to indicate that an over current condition has occurred. OC0 covers ports 0-3. The OCMAP register for the USB refers to this input signal as OC1. These signals are NOT 5V tolerant. This signal is muxed and is used by other functions.
USB_REFCLKP USB_REFCLKN	I	USB 96 MHz differential reference clock input
USB_RCOMPO	O	RCOMP OUT
USB_RCOMPI	I	RCOMP IN

14.2 Feature List

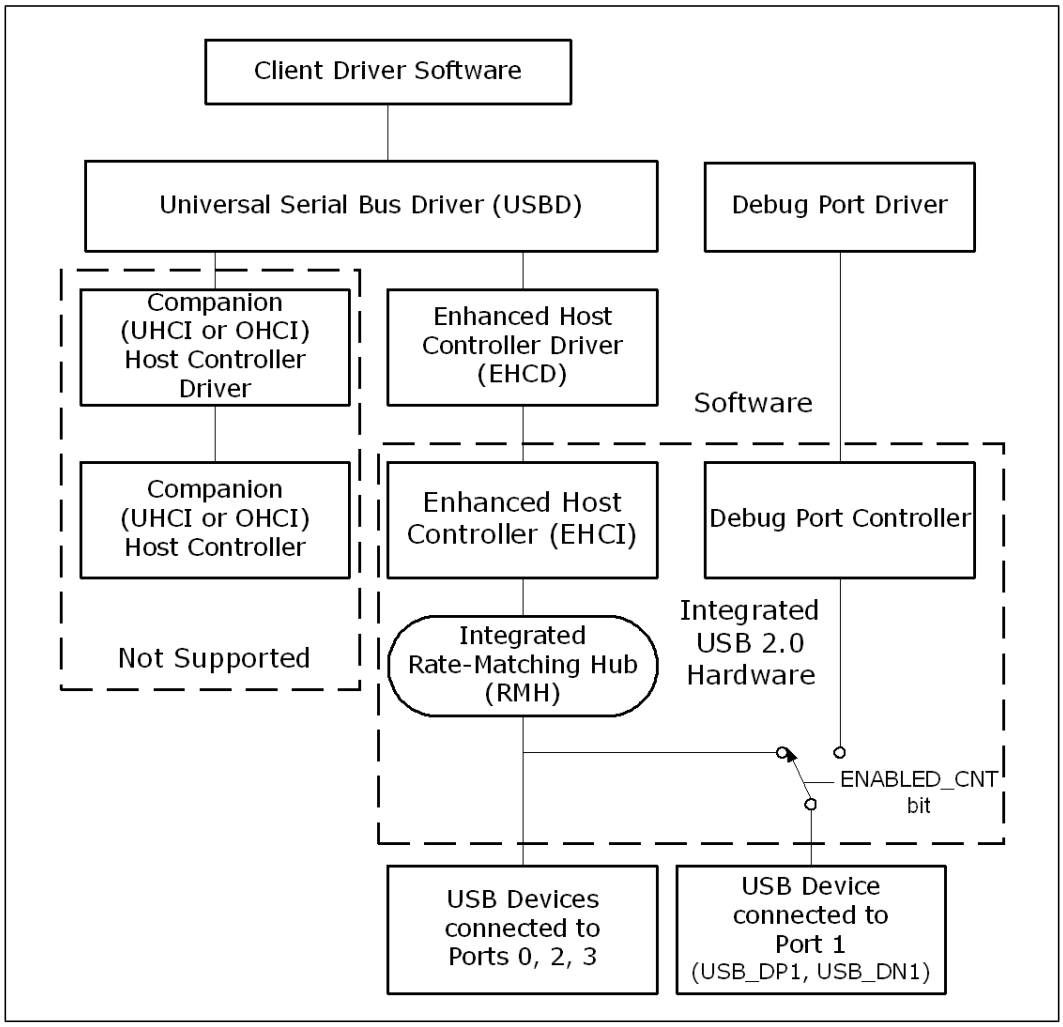
The Enhanced Host Controller (EHC) supports the following features:

- Compliant with the specification for USB 1.0, 1.1 and 2.0 (1.5 Mbps, 12 Mbps, 480 Mbps).
  - — Supports all USB transactions: bulk, isochronous, interrupt, control.
- Supports an EHCI software host controller interface.
- All ports provided by a USB Rate Matching Hub (RMH) to support Full-Speed (FS) and Low-Speed (LS) USB devices.
- One debug port having full-speed transfer rates.
- Wake-up from G2 (S5) Soft Off back to the G0 (S0) working power state.
- Asynchronous extended sleep.
- EHCI prefetch-based pause.
- Remote-suspend wake-up.
- Per-port USB disable.

14.3 Architectural Overview

The hardware appears to the system software as shown in Figure 14-2.

Figure 14-2. Software and Hardware Block Diagram



The Enhanced Host Controller (EHC) appears to the software as a PCI Express* Root Complex Integrated Endpoint.

The controller is discovered as bus 0, device 22, function 0.

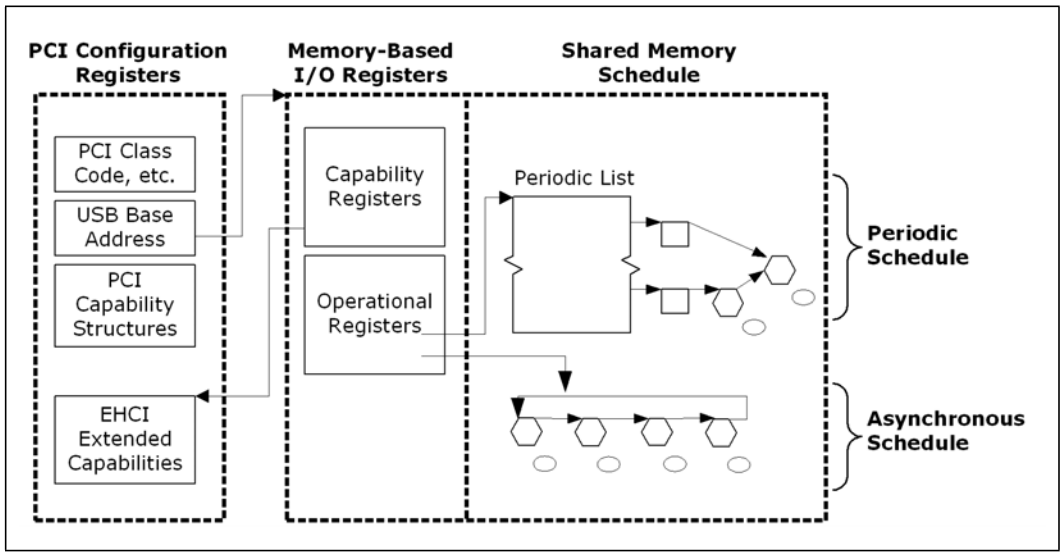
The Device ID is 0x1F2C.

The data structures in main memory are described in the rest of this section. For details, see Section 3 and Appendix B of the *Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0.

The EHC does not have a companion OHCI/UHCI host controller which use single linked-lists in memory for scheduling. The EHC separates the linked list into a Periodic List and a Asynchronous List.

The relationship of the PCI Configuration and Capabilities registers, the Memory-Mapped I/O (MMIO) registers, and schedule information are in Figure 14-3.

Figure 14-3. Software Interface Register Structure



14.3.1 PCI Configuration Registers

General information about the registers is discussed here.

Memory Base Address (base address of the MMIO registers):

- A 1-KB block of non-prefetchable, 32-bit address, memory-mapped area is requested for MMIO.

PCI Capabilities:

- PCI Power Management Capability at offset 050h.
- Debug Port Capability at offset 058h.
- Function Level Reset (FLR) Capability at offset 098h.

EHCI Capabilities:

- Legacy Support EHCI Extended Capability at offset 068h.

The Enhanced Host Controller (EHC) is required to implement the PCI Power Management registers as defined in the *PCI Bus Power Management Interface Specification*, Revision 1.2. Refer to Appendix A of the *Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0 for the EHC operational requirements for PCI Power Management.

14.3.2 Memory-Mapped I/O Registers

As shown in Figure 14-3, the EHC memory-mapped I/O space is composed of two sets of registers:

- Capability Registers
- Operational Registers

Caution: As a target of memory transactions, the EHC does not support memory transactions that are locked. Attempting to access the EHC Memory-Mapped I/O space using locked memory transactions results in undefined behavior.

Note: When the USB2 function is in the D3 PCI power state, accesses to the USB2 memory range are ignored and result in a master abort. Similarly, if the Memory Space Enable (MSE) bit is not set in the Command register in configuration space, the memory range is not decoded by the EHC. If the MSE bit is not set, then the EHC does not claim any memory accesses for the range specified in the BAR.

14.3.2.1 Host Controller Capability Registers

USB MMIO Base Address + Offset 000h to 01Fh

These registers specify the limits, restrictions, and capabilities of the EHC.

Within the Host Controller Capability Registers only the Structural Parameters register is writable. Unlike the other compatibility registers, the Structural Parameters register is implemented in the Suspend (SUS) power well and is only reset by the standard SUS power well hardware reset, not by the software Host Controller Reset (HCRESET) [bit 1 of the USB2 Command (USB2CMD) register], nor the D3-to-D0 reset.

- Offset 08h: HCCPARAMS - Host Controller Capability Parameters. This register provides general mode information that affects the generation of the data structure in memory. See Table 14-3.

Table 14-3. Host Controller Capability Parameters

Bit Field	Read/ Write	Default Value
EHCI Capabilities List	Read Only	Exists at offset 68h in the PCI configuration space.
Periodic Schedule Prefetch Capability	R/W	Supported.
Asynchronous Schedule Prefetch Capability	R/W	Supported.
Asynchronous Schedule Park Capability	Read Only	Not Supported.
Isochronous Scheduling Threshold	R/W	Host software assumes the host controller caches an isochronous data structure for an entire frame. See the *EHCI Specification* for details.
Programmable Frame List Flag	Read Only	System software must use a frame list length of 1024 elements.
64-Bit Addressing Capability	Read Only	Data structures using 64-bit address memory pointers.

14.3.2.2 Host Controller Operational Registers

USB MMIO Base Address + Offset 020h to 3FFh

These registers are divided into two sets.

1. Registers in the core power well

The first register set is at offset 20h to 3Fh. Unless otherwise noted, the core power well registers are reset by the assertion of any of the following:

- Core power well hardware reset
- Software-Controlled Host Controller Reset (HCRESET)
- D3-to-D0 reset

This first MMIO operational register set contains the USB2 Command, Status, and Interrupt Enable registers. This register set also contains the registers needed to configure and operate the Periodic Schedule and Asynchronous Schedule data structures in shared memory. The following are noteworthy:

Offset 30h: CTRLDSSEGMENT - Control Data Structure Segment Register

This register is used with the link pointers to construct 64-bit addresses to EHCI control data structures. This register is concatenated with the link pointer from either the PERIODICLISTBASE, ASYNCLISTADDR, or any control data structure link field to construct a 64-bit address.

This register allows the host software to locate all control data structures within the same 4-Gbyte memory segment.

Offset 34h: PERIODICLISTBASE - Periodic Frame List Base Address

This 32-bit register contains the beginning address of the Periodic Frame List in the system memory. Since the EHC operates in 64-bit mode (as indicated by the one in the 64-bit Addressing Capability field in the HCCPARAMS register), then the most significant 32 bits of every control data structure address comes from the CTRLDSSEGMENT register. The system software loads this register before starting the schedule execution by the Host Controller. The memory structure referenced by this physical memory pointer is assumed to be 4-KB aligned. The contents of this register are combined with the Frame Index Register (FRINDEX) to enable the Host Controller to step through the Periodic Frame List in sequence.

Offset 38h: ASYNCLISTADDR - Current Asynchronous List Address

This 32-bit register contains the address of the next asynchronous queue head to be executed. Since the EHC operates in 64-bit mode (as indicated by a one in 64-bit Addressing Capability field in the HCCPARAMS register), then the most significant 32 bits of every control data structure address comes from the CTRLDSSEGMENT register. Bits [4:0] of this register cannot be modified by the system software and always return zeros when read. The memory structure referenced by this physical memory pointer is assumed to be 32-byte aligned.

2. Registers in the Suspend (SUS) power well

The second MMIO operational register set is located at offset 60h to the end of the implemented register space. These registers are implemented in the SUS power well. Unless otherwise noted, the Core power well registers are reset by the assertion of either of the following:

- SUS power well hardware reset
- HCRESET

This second set contains the Configure Flag Register, the Status and Control Registers for the four USB Ports, and the Debug Port Registers.

14.4 Enhanced Host Controller DMA

Each DMA engine contains enough internal buffering for two maximum-sized bus transactions.

The EHC uses three sources of USB packets. In priority order for each USB microframe, these are:

1. USB 2.0 Debug Port
2. EHCI Periodic Schedule DMA Engine
3. EHCI Asynchronous Schedule DMA Engine

The EHC always performs any pending debug port transaction at the beginning of a microframe, followed by any pending periodic traffic for the current microframe. If time is left in the microframe, the EHC performs any pending asynchronous traffic until the end of the microframe (EOF1).

Note: The debug port traffic is only presented on one port (Port #1) The other ports are idle during this time.

Table 14-4. Asynchronous Schedule DMA Engine

Engine Name	Asynchronous DMA engine
Purpose	Fetch/Store Bulk and Control transfers in the main memory.
When Active	USB core is enabled, run bit is set and asynchronous schedule is enabled.
Burst Type	Two outstanding requests of 8 times 64 bytes.
Interrupts caused	Interrupts and SMIs are generated as a result of DMA transactions.

Table 14-5. Periodic Schedule DMA Engine

Engine Name	Periodic DMA engine
Purpose	Fetch/Store Interrupt and Isochronous transfers in the main memory.
When Active	USB core is enabled, run bit is set and periodic schedule is enabled.
Burst Type	Two outstanding requests of 8 times 64 bytes.
Interrupts caused	Interrupts and SMIs are generated as a result of DMA transactions.

14.5 Data Encoding and Bit Stuffing

See Chapter 8 of the *Universal Serial Bus Specification*, Revision 2.0.

14.6 Packet Formats

See Chapter 8 of the *Universal Serial Bus Specification*, Revision 2.0.

14.7 EHC Initialization

The initialization sequence expected by the EHC is described here.

The sequence begins with a complete power cycle in which the Suspend (SUS) power well and Core power well have been off.

14.7.1 Power-On

The Suspend (SUS) power well is a lower-power plane than the core power well. The SUS well is always functional when the core well is functional, but the core well is not functional when the SUS well is functional. Therefore, the SUS well reset pin (RSMRST_B) deasserts before the core well reset pin (COREPWROK) rises. The SUS well reset deasserts leaving all registers and logic in the SUS well in the default state. However, reading any registers does not occur until after the core well reset deasserts.

Note: Normally the SUS well reset only occurs when a system is unplugged (or the battery is removed). In other words, SUS well resets are not easily achieved by the software or the end user. This step typically does not occur immediately before the remaining steps.

The core well reset deasserts, leaving all registers and logic in the core well in the default state. The EHC configuration space is accessible at this point.

Note: The core well reset occurs (and typically does) without the SUS well reset asserting. This means that all of the Configure Flag and Port Status and Control bits (and any other SUS-well logic) are in any valid state at this time.

After initial power-on, whether hardware-based or via the Software-Controlled Host Controller Reset (HCRESET) bit in the USB2CMD register, all of the Operational Registers (represented as a block in Figure 14-3) are at their default values. After a hardware-based reset, only the Operational Registers not contained in the SUS power well are at their default values.

14.7.2 BIOS Initialization

The policy to disable the EHC functionality cannot be dynamic. If the EHC must be disabled in the system, the BIOS must set the corresponding function disable bit before any accesses are made to the EHC (configuration or memory space). Once set, this disable bit must remain set until a hardware reset occurs.

When the system boots the host controller is enumerated and assigned a base address for the register space, the BIOS sets the Frame Length Adjustment register located in the configuration space at bus 0, device 22 (decimal), function 0, offset 60h.

The BIOS performs a number of platform-customized steps after the Core power well has powered-up. Contact the Intel Field Representative for additional BIOS information for the SoC.

14.7.3 Port Disable Override

The BIOS and the firmware control which USB ports are usable to the end-user. They also designate which ports have non-removable devices versus which ports are exposed to the end user. The system BIOS is expected to set these values upon boot and resuming from Sx states.

14.7.4 Driver Initialization

For details, see Chapter 4 of the *Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0. Some information from that chapter is presented below.

14.7.5 EHC Resets

In addition to the standard hardware resets, portions of the EHC are reset by the software-controlled Host Controller Reset (HCRESET) bit of the USB2CMD register and the transition from the $D3_{HOT}$ device power management state to the D0 state. The effects of these resets are shown in Table 14-6.

Table 14-6. EHC Reset Types

Type of Reset	Does Reset	Does not Reset	Comments
HCRESET bit set.	Memory space registers except Structural Parameters (which is written by the BIOS).	Configuration registers.	The HCRESET must only affect registers that the EHCI driver controls. PCI Configuration space and the BIOS-programmed parameters are not reset.
D3-to-D0 Reset The software writes the Device Power State from $D3_{HOT}$ (11b) to D0 (00b).	Core power well registers (except the BIOS-programmed registers).	Suspend (SUS) power well registers. The BIOS-programmed Core power well registers.	The D3-to-D0 transition must not cause wake information (SUS well) to be lost. Also, this transition must not clear the BIOS-programmed registers because the BIOS is not invoked following the D3-to-D0 transition.

If the detailed register descriptions give exceptions to these rules, those exceptions override these rules. This summary is provided to help explain the reasons for the reset policies.

14.8 Sequence and Operating Modes

This section provides a brief overview of the USB core operation and data flow. The USB2 Enhanced Host Controller (EHC) conforms to the *Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0 and supports the programming model described in the specification. See Table 14-3.

To initialize the host controller, the software (USB driver) performs the steps required in the *EHCI Specification* and configure all the relevant registers, which include:

- Setting the periodic frame list to the PERIODICLISTBASE register, and
- The asynchronous transfer list address in the ASYNCLISTADDR register.

When all the settings are done, the controller is turned on via setting the Run/Stop bit in the Command register (USB2CMD).

At this point, the host controller is up and running and the port registers begin reporting device connects, etc. The system software enumerates a port through the reset process (where the port is in the enabled state). At this point, the port is active with Start of Frame (SOF) packets occurring at the enabled ports.

To enable USB transactions on the bus, the driver configures the Periodic Frame List and Asynchronous Transfer List data structures in the memory.

The data structures are used to communicate control, status, and data between the software and the host controller. The Periodic Frame List is an array of pointers for the periodic schedule. A sliding window on the Periodic Frame List is used. The Asynchronous Transfer List is where all the control and bulk transfers are managed and is a simple circular list of queue heads. Refer to the *EHCI Specification* for more details regarding the data structure format.

In each micro-frame, the host controller engine executes from the Periodic Schedule before executing from the Asynchronous Schedule. This engine only executes from the asynchronous schedule after it encounters the end of the periodic schedule (the software driver makes sure that the periodic schedule does not starve the asynchronous one).

The DMA engine fetches the element from the appropriate schedule and begins traversing the graph of linked schedule data structures, by issuing appropriate non-posted read requests to system memory. The DMA engine then analyzes the completion indications from the SoC memory controller.

The DMA engine starts executing the transaction on the bus once enough space is in the appropriate payload buffer (maximum-size frame for IN transactions) or a complete transaction is residing in the buffer for OUT transactions. Each DMA engine always operates on two transactions, one that is currently being executed on the bus and one that is actively being fetched from or stored to the memory. To achieve this functionality each DMA engine contains a double-buffer per direction. Each buffer is able to accommodate two maximum transactions. For Periodic DMA buffer size 1.5 KB and the Asynchronous DMA buffer size is 1 KB.

14.9 Interrupts and Error Conditions

Section 4 of the *Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0 goes into detail on the EHC interrupts and the error conditions that cause them. All error conditions that the EHC detects are reported through the EHCI Interrupt status bits. Only SoC-specific interrupt and error-reporting behavior is documented in this section. To fully comprehend the EHC interrupt and error-reporting functionality, read the EHCI Interrupts section of the specification before reading the rest of this document.

- Based on the EHC Buffer sizes and buffer management policies, the Data Buffer Error never occurs on the SoC.
- Master Abort and Target Abort responses from hub interface on EHC-initiated read packets are treated as Fatal Host Errors. The EHC halts when these conditions are encountered.
- The EHC asserts the interrupts which are based on the interrupt threshold as soon as the status for the last complete transaction in the interrupt interval has been posted in the internal write buffers. The requirement in the *Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0 (that the status is written to memory) is met internally, even though the write is not seen on DMI before the interrupt is asserted.
- Since the EHC supports the 1024-element Frame List size, the Frame List Rollover interrupt occurs every 1024 milliseconds.
- The EHC delivers interrupts to the CPU via the highly-configurable, SoC interrupt router. The software communicates with the interrupt router through the USB 8-bit Interrupt Line (ILINE_0) register in Configuration Space.
- The USB interrupt pin to IRQ mapping is platform specific. The SoC is free to choose.
- The EHC does not support Message Signaled Interrupts (MSI or MSI-X).
- The EHC does not modify the CERR count on an Interrupt IN when the Do Complete-Split execution criteria are not met.
- For complete-split transactions in the Periodic list, the Missed Microframe bit does not get set on a control-structure-fetch that fails the late-start test. If subsequent accesses to that control structure do not fail the late-start test, then the Missed Microframe bit is set and written back.

14.9.1 Aborts on USB 2.0-Initiated Memory Reads

If a read initiated by the EHC is aborted, the EHC treats the aborted read as a fatal host error. The following actions are taken when this occurs:

- The Host System Error status bit is set.
- The DMA engines are halted after completing up to one more transaction on the USB interface.
- If enabled (by the Host System Error Enable), then an interrupt is generated.
- If the status is Master Abort, then the Received Master Abort bit in configuration space is set.
- If the status is Target Abort, then the Received Target Abort bit in configuration space is set.
- If enabled (by the SERR Enable bit in the function configuration space), then the Signaled System Error bit in configuration bit is set.

14.10 Power Management

The USB controller supports the power management features described in this section.

14.10.1 Advanced Configuration and Power Interface (ACPI)

The EHC only supports the ACPI D0 (Fully On) and D3 (Fully Off) device power states and does not support the D1 and D2 intermediate power states. Concerning the device power states:

- The EHC hardware does not inherently consume any more power when in the D0 state than in the D3 state. However, the software is required to suspend or disable all ports before entering the D3 state such that the maximum power consumption is reduced.
- In the D0 state, all implemented EHC features are enabled.
- In the D3 state, accesses to the EHC memory-mapped I/O range do master abort.

Note: Since the debug port uses the same memory range, the debug port is only operational when the EHC is in the D0 state.

- In the D3 state, the EHC interrupt is not asserted for any reason. The internal Power Management Event (PME) signal indicates wake events, etc.
- When the Device Power State field is written to D0 from D3, an internal reset is generated. This is called the D3-to-D0 Reset.
- Attempts to write any other value into the Device Power State field other than 00b (D0 state) and 11b (D3 state) complete normally without changing the current value in this field.

14.10.1.1 ACPI System States

The way the EHC behavior relates to other power management states in the system (S-States) is summarized in the following list:

- The system is always in the S0 state when the EHC is in the D0 state. However, when the EHC is in the D3 state, the system is in any power management state, including S0.
- When in D0, the Prefetch-Based Pause (PBP) feature enables dynamic processor low-power states to be entered.
- The internal clock generators in the EHC are disabled when entering the S5 state (the core power turns off).
- All core well logic is reset in the S5 state.

14.10.2 Wake from System Suspend

The controller supports using the Suspend (SUS) power well to properly function in the S5 system power management states. In these Sx states, the controller SUS power well remains powered to detect wake events, such as port resume or connect/ disconnect events.

A Power Management Event (PME) is generated if the controller is armed to wake the system out of Sx back into S0.

Wakes are also a function of on-board VBUS (the voltage that the platform provides to USB-port devices along with GND) configurations. If the VBUS remains powered during Sx, USB devices generate remote walk-up signaling on the bus to wake the system from Sx states.

Upon a wake event from Sx, the host controller needs to be re-enumerated before transfers begin. USB devices are able to retain their states if VBUS continues be powered at the platform board level.

14.10.3 Asynchronous Extended Sleep

This product feature allows the Asynchronous Schedule DMA engine to remain in sleep mode for an extended period of time when certain other system conditions are met.

14.10.4 EHCI Prefetch-Based Pause

The Prefetch-Based Pause (PBP) feature works closely with the EHCI Periodic DMA Engine to enable the C2 Pop-up feature to achieve significant C3 residency even when the software has not paused or disabled the EHCI periodic schedule. PBP is completely hardware autonomous and software transparent.

This power savings is achieved by the prefetching of the periodic schedule with a series of back-to-back reads and storing information about future activity in the Host Controller, thereby creating long periods of time (up to several milliseconds) where no memory accesses (and cache snoops) occur.

To avoid race conditions with drivers that are not aware of this feature, that are updating the schedule at any given time, this feature is only enabled once the CPU(s) is in C2 or deeper. This guarantees that the data structures in memory are not modified. C2 Pop-up allows the USB controller to bring the system to C2 for a short duration and then returns the system to C3.

14.10.5 EHCI Descriptor Cache

The controller implements Asynchronous Descriptor Caching. Prefetch of the Asynchronous Schedule is performed only after CPU has entered the Cx state. When the list is empty, the controller DMA is prevented from continuously accessing memory. If the entire Asynchronous Schedule is able to be stored within the internal, 2 KB cache (2 Kbytes), the Asynchronous Schedule DMA operates out of the cache, having the benefits of asynchronous caching.

14.10.6 USB Internal Clock Shut Down

To conserve power, the EHC is able to shutdown parts of its internal clock system when certain idle conditions are met. Wake events are still detected during clock shut down.

14.10.7 Memory Latency Tolerance

Under special conditions, the platform dynamically enters deeper power savings states when all devices in the platform tolerate an established worst-case delay for access to memory. The EHC participates in this power-savings mechanism.

14.11 Security Features

14.11.1 Security Features

The USB provides a LOCK functionality, where the USB ports enable/disable register is locked by a dedicated access to the Root Complex Register Block (RCRB). Once locked, the ports enable/disable status on USB cannot be changed (i.e., the register is not writable). The lock bit is delivered as a signal from the PMC. Upon access to the lock bit in RCRB space, PMC generates a Synchronous SMI.

14.12 USB 2.0 Based Debug Port

The SoC provides the ability for the debugger software to interact with the product through one of the USB 2.0 ports (Port #1). High-level restrictions and features are:

- Must be operational before USB 2.0 drivers are loaded.
- Functions even when the port is disabled.
- The Debug Port is not used to debug an issue that requires the connection of a full-speed/low-speed device on Port #1.
- Allows normal system USB 2.0 traffic in a system that is configured to have only one USB port.
- The Debug Port device (DPD) must be high-speed capable and connect directly to Port #1. The DPD cannot be connected to Port #1 through a USB hub.
- Debug Port FIFO always makes forward progress (a bad status on the USB is presented back to the software).
- The Debug Port FIFO is only given one USB access per microframe.

The Debug port facilitates operating system and device driver debug. This port allows the software to communicate with an external console using a USB 2.0 connection. Because the interface to this link does not go through the normal USB 2.0 stack, it allows communication with the external console during cases where the operating system is not loaded, the USB 2.0 software is broken, or where the USB 2.0 software is being debugged. Specific features of this implementation of a debug port are:

- Only works with an external USB 2.0 debug device (console).
- Implemented for a specific port on the host controller.
- Operational anytime the port is not suspended AND the host controller is in D0 power state.
- Capability is interrupted when port is driving USB RESET.

14.12.1 Theory of Operation

Two operational modes for the USB debug port are:

- **Mode 1** – The USB port is in a disabled state from the viewpoint of a standard host controller driver. That is, when the Host Controller Run/Stop bit is 0. In Mode 1, the debug port controller is required to generate a keepalive packets less than 2 ms apart to keep the attached debug device from suspending. The keepalive packet is a standalone 32-bit SYNC field.
- **Mode 2** – The host controller is running (Host Controller Run/Stop bit is 1). In Mode 2, the normal transmission of SOF packets keeps the debug device from suspending.

Behavioral Rules

- In both Modes 1 and 2, the debug port controller must check for software-requested debug transactions at least every 125 microseconds.
- If the debug port is enabled by the debug driver, and the standard host controller driver resets the USB port, USB debug transactions are held off for the duration of the reset and until after the first SOF is sent.
- If the standard host controller driver suspends the USB port, then USB debug transactions are held off for the duration of the suspend/resume sequence and until after the first SOF is sent.
- The ENABLED_CNT bit in the debug register space is independent of the similar port control bit in the associated Port Status and Control register (PORTSC).

Table 14-7 shows the debug port behavior related to the state of bits in the debug registers and the bits in the associated Port Status and Control register (PORTSC).

Table 14-7. Debug Port Behavior

OWNER_CNT	ENABLED_CT	Port Enable	Run / Stop	Suspend	Debug Port Behavior
0	X	X	X	X	Debug port is not being used. Normal operation.
1	0	X	X	X	Debug port is not being used. Normal operation.
1	1	0	0	X	Debug port in Mode 1. SYNC keepalives sent plus debug traffic
1	1	0	1	X	Debug port in Mode 2. SOF (and only SOF) is sent as keepalive. Debug traffic is also sent. ***Note:*** No other normal traffic is sent out this port, because the port is not enabled.
1	1	1	0	0	Invalid. Host controller driver does not put the controller into this state (enabled, not running and not suspended).
1	1	1	0	1	Port is suspended. No debug traffic sent.
1	1	1	1	0	Debug port in Mode 2. Debug traffic is interspersed with normal traffic.
1	1	1	1	1	Port is suspended. No debug traffic sent.

14.12.1.1 OUT Transactions

An Out transaction sends data to the debug device and occurs only when the following are true:

- The debug port is enabled.
- The debug software sets the GO_CNT bit.
- The WRITE_READ#_CNT bit is set.

The sequence of the transaction is:

1. The software sets the appropriate values in the following bits:
   USB_ADDRESS_CNF
   USB_ENDPOINT_CNF
   DATA_BUFFER[63:0]
   TOKEN_PID_CNT[7:0]
   SEND_PID_CNT[15:8]
   DATA_LEN_CNT
   WRITE_READ#_CNT (*Note:* This is always 1 for OUT transactions.)
   GO_CNT (*Note:* This is always 1 to initiate the transaction.)
2. The debug port controller sends a token packet consisting of the following:
   SYNC
   TOKEN_PID_CNT field
   USB_ADDRESS_CNT field
   USB_ENDPOINT_CNT field
   5-bit CRC field
3. After sending the token packet, the debug port controller sends a data packet consisting of the following:
   SYNC
   SEND_PID_CNT field
   The number of data bytes indicated in DATA_LEN_CNT from the DATA_BUFFER
   16-bit CRC

Note: A DATA_LEN_CNT value of 0 is valid in which case no data bytes are included in the packet.

4. After sending the data packet, the controller waits for a handshake response from the debug device.

- If a handshake is received, the debug port controller:
  a. Places the received PID in the RECEIVED_PID_STS field.
  b. Resets the ERROR_GOOD#_STS bit.
  c. Sets the DONE_STS bit.
- If no handshake PID is received, the debug port controller:
  a. Sets the EXCEPTION_STS field to 001b.
  b. Sets the ERROR_GOOD#_STS bit.
  c. Sets the DONE_STS bit.

14.12.1.2 IN Transactions

An IN transaction receives data from the debug device and occurs only when the following are true:

- The debug port is enabled.
- The debug software sets the GO_CNT bit.
- The WRITE_READ#_CNT bit is reset.

The sequence of the transaction is:

1. The software sets the appropriate values in the following bits:
   USB_ADDRESS_CNF
   USB_ENDPOINT_CNF
   TOKEN_PID_CNT[7:0]
   DATA_LEN_CNT
   WRITE_READ#_CNT (*Note:* This is always 0 for IN transactions.)
   GO_CNT (*Note:* This is always 1 to initiate the transaction.)
2. The debug port controller sends a token packet consisting of the following:
   SYNC
   TOKEN_PID_CNT field
   USB_ADDRESS_CNT field
   USB_ENDPOINT_CNT field
   5-bit CRC field
3. After sending the token packet, the debug port controller waits for a response from the debug device.
   If a response is received:
   a. The received PID is placed into the RECEIVED_PID_STS field.
   b. Any subsequent bytes are placed into the DATA_BUFFER.
   c. The DATA_LEN_CNT field is updated to show the number of bytes that were received after the PID.
4. If the valid packet was received from the device that was one byte in length (indicating it was a handshake packet), then the debug port controller:
   a. Resets the ERROR_GOOD#_STS bit.
   b. Sets the DONE_STS bit.
5. If valid packet was received from the device that was more than one byte in length (indicating it was a data packet), then the debug port controller:
   a. Transmits an ACK handshake packet.
   b. Resets the ERROR_GOOD#_STS bit.
   c. Sets the DONE_STS bit.
6. If no valid packet is received, then the debug port controller:
   a. Sets the EXCEPTION_STS field to 001b.
   b. Sets the ERROR_GOOD#_STS bit.
   c. Sets the DONE_STS bit.

14.12.1.3 Debug Software

14.12.1.3.1 Enabling the Debug Port

Two mutually exclusive conditions that the debug software must address as part of its startup processing are:

- The EHCI has been initialized by the system software.
- The EHCI has not been initialized by the system software.

The debug software determines the current initialized state of the EHCI by examining the configure flag in the EHCI USB 2.0 Command register. If this flag is set, then the system software has initialized the EHCI. Otherwise, the EHCI is not considered initialized. The debug software initializes the debug port registers depending on the state the EHCI. However, before this is accomplished, the debug software must determine which root USB port is designated as the debug port.

14.12.1.3.2 Determining the Debug Port

The debug software easily determines which USB root port has been designated as the debug port by examining bits [20:23] of the EHCI Host Controller Structural Parameters register. For the SoC, this 4-bit field is hardwired as 2h indicating that the Debug Port is on the second port on the EHC which is Port #1.

14.12.1.3.3 Debug Software Startup with Non-Initialized EHCI

The debug software attempts to use the debug port if after setting the OWNER_CNT bit, the Current Connect Status bit in the appropriate (see Determining the Debug Port) PORTSC register is set. If the Current Connect Status bit is not set, then the debug software chooses to terminate or to wait until a device is connected.

If a device is connected to the port, then the debug software must reset/enable the port. This is done by setting and then clearing the Port Reset bit in the PORTSC register. To ensure a successful reset, the debug software waits at least 50 ms before clearing the Port Reset bit. Due to delays, this bit does not change to 0 immediately; reset is complete when this bit reads as 0. The software must not continue until this bit reads 0.

If a high-speed device is attached, the EHCI automatically sets the Port Enabled/Disabled bit in the PORTSC register and the debug software proceeds. The debug software sets the ENABLED_CNT bit in the Debug Port Control/Status register (DP_CTRLSTS), and then resets (clears) the Port Enabled/Disabled bit in the PORTSC register (so that the system host controller driver does not recognize an enabled port when it is first loaded).

14.12.1.3.4 Debug Software Startup with Initialized EHCI

The debug software attempts to use the debug port if the Current Connect Status bit in the appropriate (see Determining the Debug Port) PORTSC register is set. If the Current Connect Status bit is not set, then the debug software chooses to terminate or chooses to wait until a device is connected.

If a device is connected, then the debug software must set the OWNER_CNT bit and then the ENABLED_CNT bit in the Debug Port Control/Status register (DP_CTRLSTS).

14.12.1.3.5 Determining Debug Peripheral Presence

After enabling the debug port functionality, the debug software determines if a debug peripheral is attached by attempting to send data to the debug peripheral. If all attempts result in an error (exception bits in the Debug Port Control/Status register (DP_CTRLSTS) indicates a transaction error), then the attached device is not a debug peripheral. If the debug port peripheral is not present, then the debug software chooses to terminate or chooses to wait until a debug peripheral is connected.

14.13 USB Over-Current Protection

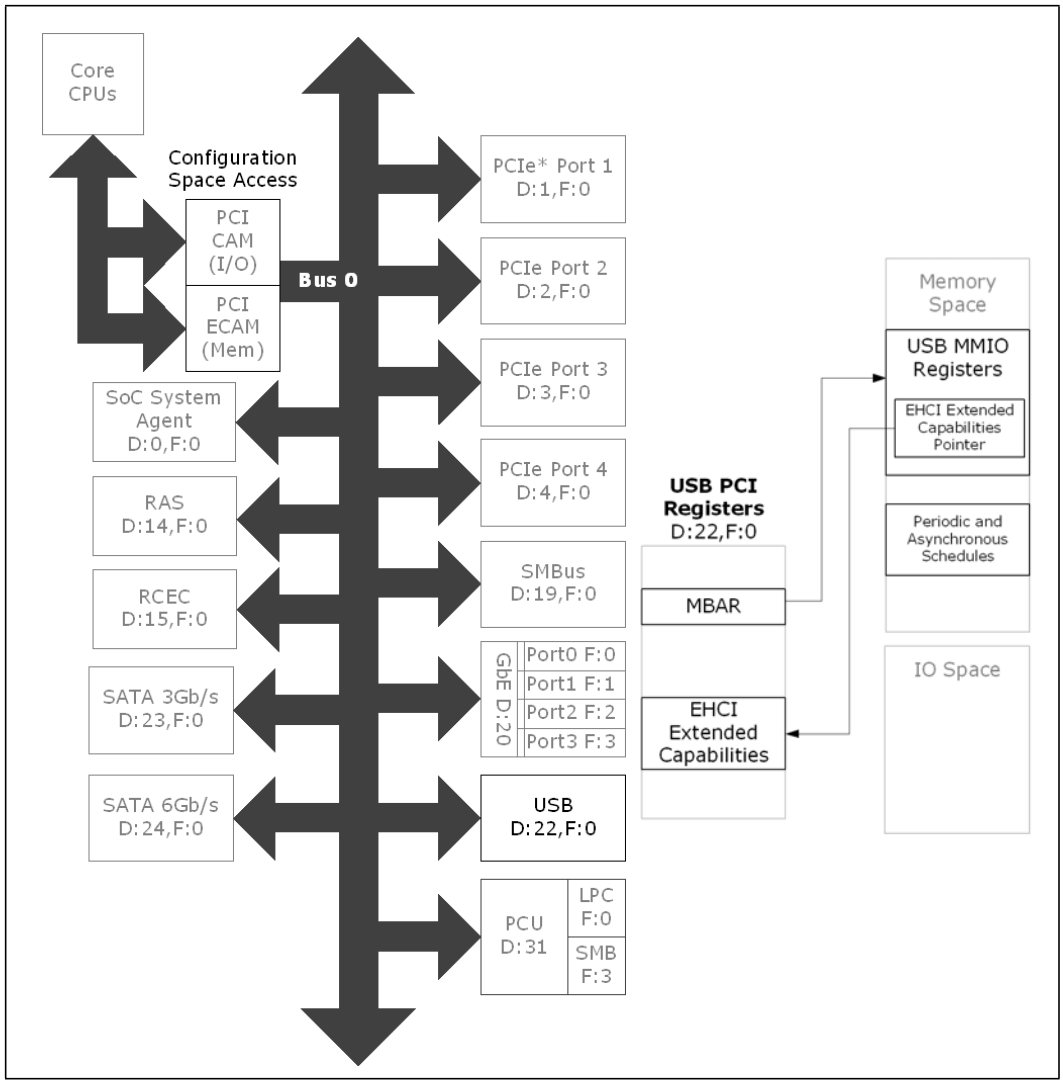
The SoC provides one over-current indicator pin, USB_OC0_B, that is shared across the four USB ports. The pin is an active-low, 3.3V signal and is not 5V tolerant. It is important that:

- All the USB ports routed out of the package must have Over-Current (OC) protection. The system BIOS ensures all used ports have OC protection.
- Unused USB Ports on the system (not routed out from the package) do not have OC pins assigned to them.

14.14 Register Map

Figure 14-4 shows the C2xx0USB controller registers from a system viewpoint.

Figure 14-4. USB Register Map



14.14.1 PCI Configuration and Capabilities

Table 14-8. USB 2.0 Controller PCI Configuration and Capabilities Register Map

CFG Address	Name	Description
0x00	VID_DID	Vendor ID and Device ID
0x04	CMD_STS	Command and Device Status
0x08	RID_PI_CC	Revision ID and Programming Interface and Sub/Base Class Code
0x0C	RSVD_MLT_HT	Reserved and Master Latency Timer and Header Type
0x10	MBAR	Memory Base Address
0x2C	SSVID_SSID	USB2 Subsystem Vendor ID and USB2 Subsystem ID
0x34	CAP_PTR_RSVD	Capabilities Pointer and Reserved
0x3C	ILINE_IPIN_RSVD	Interrupt Line and Interrupt Pin and Reserved
0x44	IHFCLK	Intel-Specific High Precision Frame Clock
0x48	IHFCLKC	Intel-Specific High Precision Frame Clock Capture
0x50	PM_CID_NEXT_CAP	PCI Power Management Capability ID-Next Item Pointer #1-PM Capabilities
0x54	PM_CS	Power Management Control/Status
0x58	DP_CID_NEXT_BASE	Debug Port Capability ID-Next Item Pointer #2-Debug Port Base Offset
0x60	SBRN_FLA_PWC	Serial Bus Release Number-Frame Length Adjustment-Port Wake Capability
0x68	ULSEC	USB2 Legacy Support Extended Capability
0x6C	ULSCS	USB2 Legacy Support Control/Status
0x74	OCMAP	Overcurrent Mapping
0x7E	RMHWKCTL	RMH Wake Control
0x98	FLR_CID_NEXT_MISC	Function Level Reset capability ID-Next Capability Pointer-Length-Version
0x9C	FLR_CTL_STS_RSVD	Function Level Reset (FLR) Control Register-Status Register-Reserved
0xF8	MANID	Manufacturer ID

14.14.2 MMIO Registers

Table 14-9. USB 2.0 Controller MMIO Register Map

MEM Address	Name	Description
0x00	CAP_HCIV	Capability Registers Length and HC Interface Version Number
0x04	HCSPARAMS	Host Controller Structural Parameters
0x08	HCCPARAMS	Host Controller Capability Parameters
0x0C	CPRD	Companion Port Route Description
0x20	USB2CMD	USB2 Command Register
0x24	USB2STS	USB2 Status
0x28	USB2INTR	USB2 Interrupt Enable
0x2C	FRINDEX	Frame Index
0x30	CTRLDSSEGMENT	Control Data Structure Segment Register
0x34	PERIODICLISTBASE	Periodic Frame List Base Address
0x38	ASYNCLISTADDR	Current Asynchronous List Address
0x60	CONFIGFLAG	Configure Flag Register
0x64	PORTSC1	Port Status and Control
0x68	PORTSC2	Port Status and Control
0x6C	PORTSC3	Port Status and Control
0x70	PORTSC4	Port Status and Control
0xA0	DP_CTRLSTS	Debug Port Control/Status Register
0xA4	DP_USB_PIDs	USB PIDs Register
0xA8	DP_DATA_BUF_B	Debug Port Data Buffer Bytes [7:0]
0xB0	DP_CFG	Debug Port Configuration Register
0xF0	RMHPORTSTS1	RMH Port Status Register 1

§ §

15 SMBus 2.0 Unit 1 - Host

The SMBus Message Transport (SMT) controller provides a mechanism whereby the SoC sends and receives Out-Of-Band manageability messages over the SMBus to Managed Devices or to Management Controllers. This allows it to participate in manageability services with Intel and third-party devices like embedded controllers, sensors, and other devices through the SMBus interface.

Figure 15-1. SMBus Host Covered in This Chapter

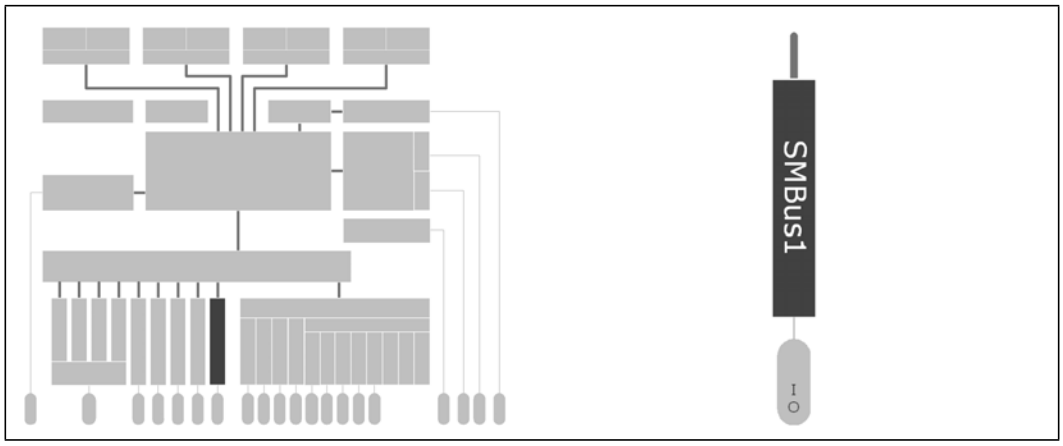


Table 15-1. References

Reference	Revision	Date	Document Title
SMBus	2.0	August 3, 2000	*System Management Bus (SMBus) Specification,* Version 2.0
IPMI	1.0	Feb. 12, 2004	*Intelligent Platform Management Interface (IPMI) Specification,* Version 2.0
ASF Specification (DSP0136)	-	April 23, 2003	*Alert Standard Format (ASF) Specification,* Version 2.0.0
MCTP Specification (DSP0236)	-	July 28, 2009	*Management Component Transport Protocol (MCTP) Base Specification,* Version 1.0.0

15.1 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

Table 15-2. Signal Names

Signal Name	Direction Type	Description
SMB_CLK1	I/OD	SMBus Clock (SMBCLK) *This signal is muxed with GPIOS_12 and is used by other functions.*
SMB_DATA1	I/OD	SMBus Data (SMBDAT) *This signal is muxed with GPIOS_11 and is used by other functions.*

The optional SMBus 2.0 signals, SMBALERT# and SMBSUS#, are not supported on this controller interface.

15.2 Features

- Operates as an SMBus master or target
- Supports ARP in master or slave mode
- PEC is enabled for SMBus transactions
- Compatible with certain I^2C master (all commands except read-then-write) and slave (only MTx-to-SRx) modes
- Status and errors are communicated by polling or interrupts
- Supports INTx or MSI

15.3 Architectural Overview

The SMBus Message Transport (SMT) controller has three regions in Configuration Space. These are discovered by software in the configuration space as bus 0, device 19 (decimal), function (0). The address offsets and capability IDs of the contents of these regions are:

1. PCI Standard Header
   - Type 0
2. PCI Capabilities List
   - 40h: PCI Express* - Capability ID = 10h
   - 80h: PCI Power Management - Capability ID = 01h
   - 8Ch: Message Signaled Interrupts (MSI) - Capability ID = 05h
   - Various implementation-specific and Intel-reserved registers
3. PCI Express Extended Capabilities List
   - 100h: Advanced Error Reporting (AER) - Extended Capability ID = 0001h

The SMT controller operation is DMA-based in which descriptors and data are exchanged between the SoC hardware and the firmware through system memory. This DMA mode is available for the SMT acting both as master and target.

As transport layer functionality, SMT transfers messages between the devices on the SMBus segment (to which it is physically connected) and the SoC firmware. The SMT hardware is physically located within the SoC and resides within its PCIe space. In the PCIe hierarchy, SMT is bus 0, device 19 (decimal), function 0.

Table 15-3 summarizes which SMBus ARP, SMBus, and I^{2}C protocols are supported by the SMT controller.

Table 15-3. List of Supported SMBus ARP, SMBus, and I^{2}C Protocols

Protocol	Supported	
SMBus ARP Commands	**Sent as ARP Master**	**Received as ARP Slave**
Prepare to ARP	Yes	
Reset Device *general and directed*		
Get UDID *general and directed*		
Assign Address		
SMBus ARP Commands	**Sent as ARP Slave**	**Received as ARP Master**
Notify ARP Master	Yes	
SMBus Commands	**Sent as Master**	**Received as Slave**
Quick Command	Yes	Yes
Send Byte	Yes	Yes
Receive Byte	Yes	No
Write Byte/Word	Yes	Yes
Read Byte/Word	Yes	No
Process Call	Yes	No
Block Write	Yes	Yes
Block Read	Yes	Yes
Block Write-Block Read Process Call	Yes	No
SMBus Host Notify	Yes	Yes
I^{2}C Commands	**Sent as Master**	**Received as Slave**
Master-TX writes Slave-RX (MTx-to-SRx) *no direction change*	Yes	Yes
Master-RX reads Slave-TX (STx-to-MRx)	Yes	No
Combined format, Write-then-Read *direction change after initial Write*	Yes	No
Combined format, Read-then-Write *direction change after initial Read*	No	No

15.4 Controller Characteristics and Operation

15.4.1 Electrical

SMBus physical segments must comply with the high-power DC electrical specifications as defined in the *System Management Bus (SMBus) Specification,* Version 2.0.

The maximum capacitance of each SMBus signal pin is 40 pF, which includes the sum of all device capacitance loads and capacitance of trace length. The absolute value of the total leakage current for an SMBus physical segment, source, and/or sink, must be less than 200 µA measured at 0.1 x Vcc and 0.9 x Vcc.

15.4.2 SMBus Behavior on PCIe Reset

When power is applied to an SMBus device, it performs default initialization of internal state as specified in the *System Management Bus (SMBus) Specification,* Version 2.0 SMBus device interface logic is not affected by PERST#. This normally allows the SMBus to support communications when the PCIe* interface cannot.

15.4.3 Addressing and Configuration

An Address Resolution Protocol (ARP) is defined in the *System Management Bus (SMBus) Specification,* Version 2.0 as assigning slave addresses to SMBus devices. It is required that systems that connect the SMBus to PCIe slots implement the ARP for assignment of SMBus slave addresses to SMBus interface devices on PCIe add-in cards. The system must execute the ARP on a logical SMBus whenever any PCIe device in an individual slot associated with the logical SMBus exits the $D3_{COLD}$ state. Before executing the ARP, the system must ensure that all ARP-capable SMBus interface devices are returned to their default address state.

15.4.3.1 ARP Nomenclature

The following are some definitions pertaining to SMBus ARP.

Table 15-4. ARP Nomenclature

Term	Definition
Address Resolution Protocol (ARP)	A protocol by which SMBus devices with assignable addresses on the bus are enumerated and assigned non-conflicting slave addresses.
Address Resolved (AR) flag	A flag bit or state internal to the SoC that indicates whether or not the device slave address has been resolved by the ARP Master.
Address Valid (AV) flag	A flag bit or state internal to SoC that indicates whether or not the device slave address is valid. This bit must be non-volatile for devices that support the Persistent Slave Address.
ARP Master	The SMBus master executes the ARP and assigning addresses to ARP-capable slave devices. The SMBus Host usually is the ARP Master, but under some circumstances another SMBus master assumes the role. Only one active ARP Master exists at any time.
Persistent Slave Address (PSA)	An assigned slave address that is retained through the loss of device power.
SMBus Device Default Address	The address all ARP-capable slave device must respond to. After a slave address has been assigned a device must still respond to commands at the SMBus Device Default Address for ARP management. This address is fixed at 1100 001.
SMBus Host	A specialized SMBus Master that provides the main interface to the system CPU. It must be a master-slave and must support the SMBus host notify protocol. At most, one host is in a system.
Unique Device Identifier (UDID)	A 128-bit value that a device uses during the ARP process to uniquely identify itself.

Table 15-5. Device Decodes of AV and AR Flags

Address Valid (AV)	Address Resolved (AR)	Meaning
Cleared	Cleared	The device does not have a valid slave address and participates in the ARP process. This is the POR state for a device that does not support the PSA (persistent slave address) or if it does it has not previously been assigned a slave address.
Cleared	Set	ILLEGAL STATE
Set	Cleared	The device has a valid slave address but must still participate in the ARP process.
Set	Set	The device has a valid slave address that has been resolved by the ARP Master. The device does not respond to the Get UDID (general) command. However, it subsequently receives an Assign Address command and changed its slave address accordingly.

15.4.3.2 Unique Device Identifier (UDID) Format

The UDID is a 128-bit value that a device uses during the ARP process to uniquely identify itself.

Table 15-6. UDID Format

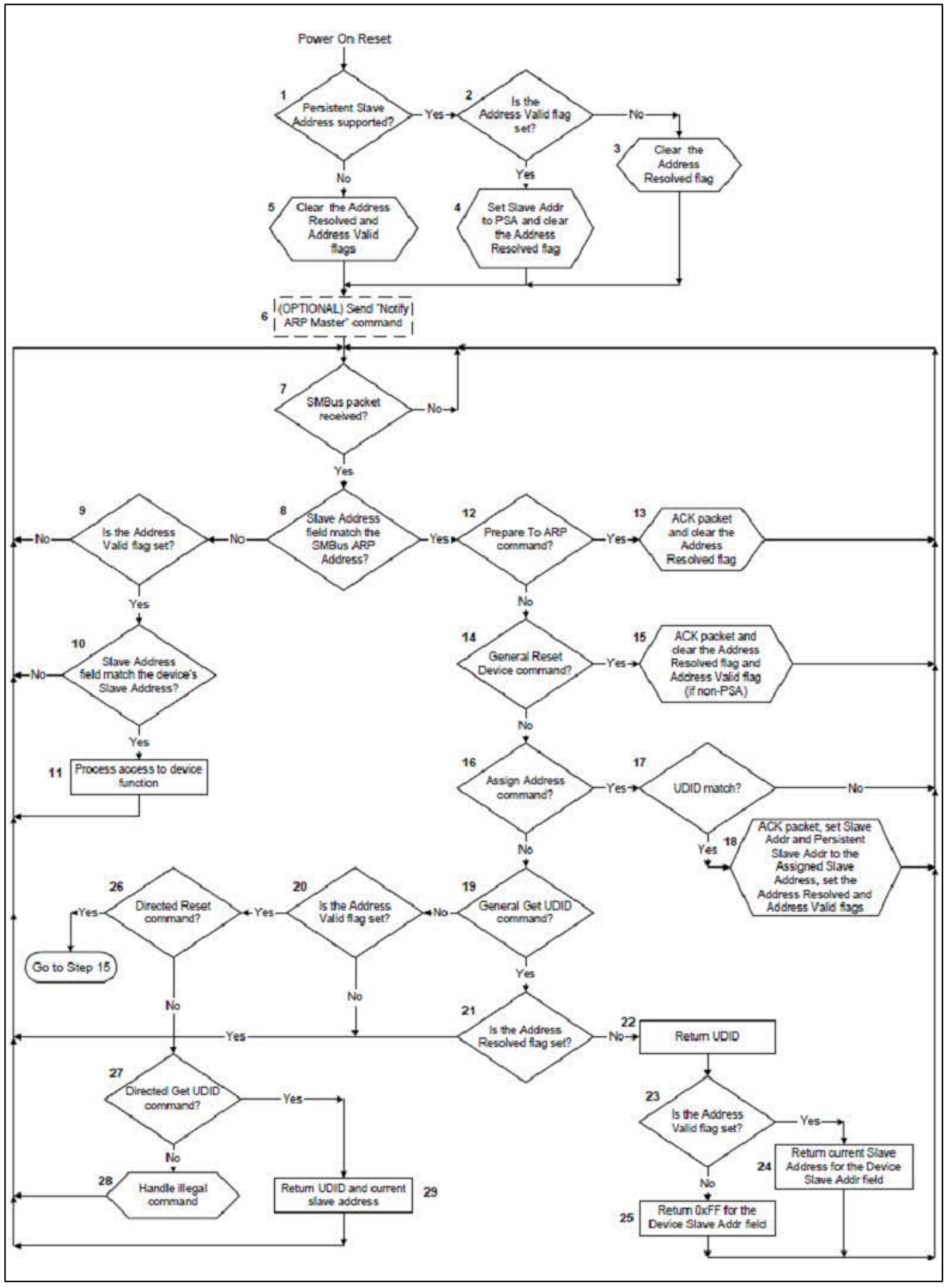
Size	Field	Comments
8 bits (MSB)	Device Capabilities	Includes selection of fixed or dynamic ARP address
8 bits	Version/Revision	
16 bits	Vendor ID	
16 bits	Device ID	
16 bits	Interface	
16 bits	Subsystem Vendor ID	
16 bits	Subsystem Device ID	
32 bits (LSB)	Vendor Specific ID	Includes bits to uniquely identify each device UDIDs

The vendor-specific field provides a unique ID for functionally equivalent devices. This field is for devices that otherwise return identical UDIDs for the purpose of address assignment. A unique ID in this field is required since this device supports an assigned slave address. For a pre-assigned unique ID, at least 24 bits must be unique; however the full 32 bits is recommended. Uniqueness is important to guarantee that two like devices are identified discretely.

The Vendor Specific ID (VSID) fields of the UDID0 Data Register (UDID0) and UDID1 Data Register (UDID1) are located in the SMT controller MMIO.

15.4.3.3 ARP Slave Behavior

Figure 15-2. ARP-Capable (Slave) Device Behavior Flow Diagram



With reference to Table 15-7 the ARP slave operates as follows (steps which are shaded are the responsibility of the firmware alone or of the firmware to trigger the hardware).

Table 15-7. ARP Slave Operations (Sheet 1 of 3)

Step		Description
0. . . Perform ARP slave initialization.		• [FW[1]] See Section 15.4.3.5, "ARP Initialization Flow" on page 302
1.	After exiting the power-on-reset state, a device that supports PSA goes to step 2 to verify if its slave address is valid. If the device does not support PSA, it proceeds to step 5.	• [FW] SoC does not support PSA; jump past PSA-related steps to step 5
2.	A device supporting PSA must check its Address Valid flag which is non-volatile. If that flag is set then it has previously received an assigned slave address; proceed to step 4. If the Address Valid flag is cleared then it must proceed to step 3.	• N/A
3.	Although the device supports PSA the value is currently invalid. The device must clear the Address Resolved flag to indicate that it has not had its slave address assigned. Proceed to step 6.	• N/A
4.	The device has a valid PSA so it assumes that slave address for now. However, this address has not been resolved by the ARP Master so the device must clear its Address Resolved flag. Proceed to step 6.	• N/A
5.	The device does not support PSA so it must clear its Address Valid and Address Resolved flags. Proceed to step 6.	• [FW] Deassert SMTARPCTRL.AV* • [FW] Deassert SMTARPCTRL.AR*
6.	If supported, the device masters the SMBus and sends the Notify ARP Master command. This informs the ARP Master that a new device is present. Proceed to step 7.	• [FW] Determine if supported; check SMTARPCTRL.NOTIFYENB • [FW and HW[2]] Initiate Master Transaction (see Section 15.4.7.4, "Master Transactions Flow" on page 314)
7.	The device waits for an SMBus packet.	• [FW and HW] Wait
8.	Upon receipt of an SMBus packet the device must first check the received slave address against the SMBus Device Default Address. If a match is present, then it proceeds to step 12, otherwise it proceeds to step 9.	• [HW] Match the default address • [HW] ACK the transaction if received slave address matches Device Default Address
9.	The received address is not the SMBus Device Default Address so the packet is potentially addressed to one of the device core functions. The device must check its Address Valid bits to determine whether or not to respond. If any of the Address Valid bits are set then it proceeds to step 10, otherwise it must return to step 7 and wait for another SMBus packet.	• [HW] Check if any SMTARPCTRL.AV*n* are asserted
10.	Since the device has a valid slave address it must compare the received slave address to its internal slave address. If a match is present, then it proceeds to step 11, otherwise it must return to step 7 and wait for another SMBus packet.	• [HW] Match any of the internal slave addresses • [HW] ACK the transaction if received slave address matches **any** target address **and** that target Address Valid bit (SMTARPCTRL.AV*n*) is asserted
11.	The device has received a packet addressed to a core function so it acknowledges the packet and processes it accordingly. Proceed to step 7 and wait for another SMBus packet.	• [FW and HW] Initiate Target Transaction (see Section 15.4.8.4, "Target Memory Buffer Hardware-Firmware Flow" on page 327)
12.	The device detected a packet addressed to the SMBus Device Default Address. It must check the command field to determine if this is the Prepare To ARP command. If so, then it proceeds to step 13, otherwise it proceeds to step 14.	• [FW] Inspects command field and determines how to proceed
13.	Upon receipt of the Prepare To ARP command the device must acknowledge the packet and make sure its Address Resolved flag is clear to participate in the ARP process. Proceed to step 7 and wait for another SMBus packet.	• [FW] Deassert SMTARPCTRL.AR*
14.	The device checks the command field to verify if the Reset Device command was issued. If so, then it proceeds to step 15, otherwise it proceeds to step 16.	• [FW] Inspects command field and determines how to proceed

Table 15-7. ARP Slave Operations (Sheet 2 of 3)

Step		Description
15.	Upon receipt of the Reset Device command the device must acknowledge the packet and make sure its Address Valid (if non-PSA) and Address Resolved flags are cleared. This allows the ARP Master to re-assign all device addresses without cycling power. Proceed to step 7 and wait for another SMBus packet.	• [FW] Deassert SMTARPCTRL.AV* • [FW] Deassert SMTARPCTRL.AR*
16.	The device checks the command field to verify if the Assign Address command was issued. If so, then it proceeds to step 17, otherwise it proceeds to step 19.	• [FW] Inspects command field and determines how to proceed
17.	Upon receipt of the Assign Address command the device must compare its UDID to the one it is receiving. If any byte does not match then it must not acknowledge that byte or subsequent ones. If all bytes in the UDID compare then the device proceeds to step 18, otherwise it must return to step 7 and wait for another SMBus packet.	• [HW] Check command field **only if** received slave address matches Device Default Address • [HW] If the Assign Address command, then verify each byte of received UDID against all SMT UDIDs. • [HW] NACK the transaction if received UDID fails to match any SMT UDID ***Note:*** All ARP protocols are directed only to the Device Default Address. ***Note:*** SMT must respond to this command even if its SMTARPCTRL.AR flag is set (i.e., ARP Master overwrites a valid and resolved target address). ***Note:*** This also implies the hardware cannot detect an ARP-pending flag to selectively monitor the command field for Assign Address command. ***Note:*** If UDID is NACKed, the hardware is not required to notify the firmware.
18.	Since the UDID matched, the device must assume the received slave address and update its PSA, if supported. The device must set its Address Valid and Address Resolved flags; this indicates it no longer responds to the Get UDID command unless it receives the Prepare To ARP or Reset Device commands or is power cycled. Proceed to step 7 and wait for another SMBus packet.	• [FW] Assign the received slave address to the matched UDID • [FW] Assert SMTARPCTRL.AV*n* • [FW] Assert SMTARPCTRL.AR*n*
19.	The device checks the command field to verify if the Get UDID command was issued. If so, then it proceeds to step 21, otherwise it proceeds to step 20.	• [HW] Inspects command field and determines how to proceed
20.	The device is receiving a directed command. Directed commands must be acknowledged only by slaves with a valid address. If the address is not valid then ignore the packet and return to step 7 and wait for another SMBus packet. If the address is valid then proceed to step 26.	• [HW] Inspects Address Valid flag
21.	Upon receipt of the Get UDID command, the device must check its Address Resolved flag to determine whether or not it participates in the ARP process. If set then its address has already been resolved by the ARP Master so the device proceeds to step 7 to wait for another SMBus packet. If the AR flag is cleared then the device proceeds to step 22.	• [HW] Inspects Address Resolved flag and determines how to proceed
22.	The device returns its UDID and monitors the SMBus data line for collisions. If a collision is detected at any time the device must stop transmitting and proceed to step 7 and wait for another SMBus packet. If no collisions were detected then proceed to step 23.	• [HW] UDID is returned as read data to master
23.	The device must now check its Address Valid flag to determine what value to return for the Device Slave Address field. If the AV flag is set then it proceeds to step 24, otherwise it proceeds to step 25.	• [HW] Inspects Address Valid flag and determines how to proceed
24.	The current slave address is valid so the device returns this for the Device Slave Address field (with bit 0 set) and monitors the SMBus data line for collisions (i.e., another device driving a 0 when this device is driving a 1). Proceed to step 7 and wait for another SMBus packet.	• [HW] UDID and current slave address are returned as read data to master (as per generic block-read payload)
25.	The current slave address is invalid so the device returns a value of 0xFF and monitors the SMBus data line for collisions. If the ARP Master receives the 0xFF value it knows that the device requires address assignment. Proceed to step 7 and wait for another SMBus packet.	• [HW] UDID and 0xFF address are returned as read data to master (as per generic block-read payload)

Table 15-7. ARP Slave Operations (Sheet 3 of 3)

Step		Description
26.	Is this a directed Reset Device command? If so then proceed to step 15. Otherwise proceed to step 27.	• [FW] Inspects command field and determines how to proceed
27.	Is this the Get UDID (directed) command? If so then proceed to step 29. Return the UDID information. If not, then proceed to step 28.	• [HW] Inspects command field and determines how to proceed
28.	The device has not received a valid command so it must handle the illegal command in accordance with SMBus rules for error handling. Proceed to step 7 and wait for another SMBus packet.	• [FW] Perform error handling
29.	Return the UDID information and current slave address, then proceed to step 7 and wait for another SMBus packet.	• [HW] UDID and current slave address are returned as read data to master (as per generic block-read payload)

1. FW means Firmware.
2. HW means Hardware.

Table 15-8 identifies the hardware pattern-matching which is required to decode ARP and ordinary SMBus protocols. The UDID matching applies only to Assign Address protocol.

Since ARP protocols are typically writes to the ARP slave by the ARP host, the R/W# bit (LSB) of the first byte is usually 0; however, the general and directed forms of Get UDID require the ARP slave to return its UDID to the ARP host, such that the LSB of the first byte after the repeated-start is 1. During either Get UDID protocol, the hardware matching must detect the repeated-start and subsequently match a slave address of C3h.

Likewise, most ordinary SMBus protocols begin with a write to the slave/target by the master. However, since certain protocols begin with a read to the slave, the hardware matching treats the LSB of the first byte as a (don’t care). For those protocols which require the slave to return data to the master, the LSB of the first byte after the repeated-start is 1. Therefore, the hardware matching must detect the repeated-start and subsequently match a slave address with an appended 1 LSB.

Table 15-8. Hardware Decoding of ARP, SMBus, and I²C Target Transactions

Byte 1 Slave Address[1]	Byte 2 ARP Command	Bytes 4-19 UDID	Hardware-Decoded Protocol0
ARP protocols applicable to all SMT UDIDs			
C2h	01h	n/a	Prepare to ARP
C2h	02h	n/a	Reset Device (general)
C2h	03h	n/a	Get UDID (general)
C3h[2]	n/a	n/a	Get UDID (general) **—after repeated start**
ARP protocols specific to only matched UDID (128 bits)			
C2h	04h	Matches an SMT UDID	Assign Address
ARP protocols specific to only matched slave address (7 bits)			
C2h	{target address, 1}	n/a	Get UDID (directed)
C3h[2]	n/a	n/a	Get UDID (directed) **—after repeated start**
C2h	{target address, 0}	n/a	Reset Device (directed)
10h	C2h	n/a	Notify ARP master
10h	!= C2h[3]	n/a	SMBus Host Notify
Ordinary SMBus (and I²C) protocol specific to only matched slave address (7 bits)			
{slave address, x}	n/a	n/a	Ordinary SMBus transaction. LSB = x to be protocol agnostic • Quick Command • Send Byte • Write Byte/Word • Block Write • Block Read
{slave address, **1**}[2]	n/a	n/a	Ordinary SMBus transaction. LSB = 1 **—after repeated start** • Block Read

1. Hardware checking of LSB during start address cycle is controlled by SUSCHKB.IRWST. (Applies to SMBus target addresses, C2h, and 10h.) See also Section 15.4.8.5, "Target Flow" on page 330 and Table 15-20 for more implications to the firmware and the hardware regarding IRWST.
2. Hardware checking of LSB during repeated-start address cycle is controlled by SUSCHKB.IRWRST. (Applies to SMBus target addresses and C2h.)
3. For a transaction directed to 10h if the command byte is not C2h then the transaction is SMBus Host Notify and the command code format is {initiator slave address, 0}.

Certain SMBus protocols are disallowed in target mode since they violate the hardware-firmware descriptor mechanism. First, the descriptor is passed to the firmware only after the transaction stop; this architecture does not permit the firmware to return a payload within the same transaction. Second, waiting for a descriptor from the firmware requires the hardware to perform excessive SMBus clock-stretching.

Table 15-9 captures the responses of the hardware/firmware under the various combinations of ARP protocol, address and UDID fields, and ARP status flags. The states mentioned refer to Table 15-9.

Table 15-9. Hardware/Firmware Response to SMBus and ARP Protocols

Received ARP Protocol	AV Flag	AR Flag	ARP States (1-8,...)	SoC Hardware and Firmware Response
Ordinary SMBus protocol				
Ordinary SMBus protocol	T	n/a	9,10,11	Normal hardware-firmware flow
SMBus block-read	T	n/a	9,10,11	The hardware returns the specified number of bytes from the hardware buffers.
ARP protocol				
Prepare to ARP	n/a	n/a	12,13	Normal hardware-firmware flow
Reset Device (general)	n/a	n/a	12,14,15	Normal hardware-firmware flow
Assign Address	n/a	n/a	12,14,16, 17,18	Normal hardware-firmware flow
Get UDID (general)	n/a	T	12,14,16, 19,21	The hardware does NACK the ARP command byte to indicate it has a valid assigned slave address (ARP is complete).
Get UDID (general)	T	F	12,14,16, 19,21,22, 23,24	The hardware returns its UDID and corresponding Slave Address.
Get UDID (general)	F	F	12,14,16, 19,21,22, 23,25	The hardware returns its UDID and 0xFF for its address.
Reset Device (directed)	T	n/a	12,14,15, 19,20,26, 15	Normal hardware-firmware flow
Get UDID (directed)	T	n/a	12,14,16, 19,20,26, 27,29	The hardware returns its UDID and corresponding Slave Address.
None of the above ARP protocols	n/a	n/a	12,14,16, 19,20,26, 27,28	Normal hardware-firmware flow (error handling situation)

The hardware will ACK all valid transactions above. An exception is when processing the Assign Address protocol, if the received UDID does not match any internal UDID the hardware begins NACKing after the first unmatched byte. Another exception is that if the Address Resolved flag is asserted and a Get UDID (general) command is received, the command byte is NACKed. Finally, a PEC failure causes a transaction to be NACKed.

In both directed protocols, Get UDID (directed) and Reset Device (directed), the second byte, or ARP command byte, consists of the target slave address with an appended 0 or 1 to denote a Reset Device or Get UDID protocol, respectively. Therefore, the hardware uses address bits [7:1] for address comparison and LSB bit 0 to determine the protocol.

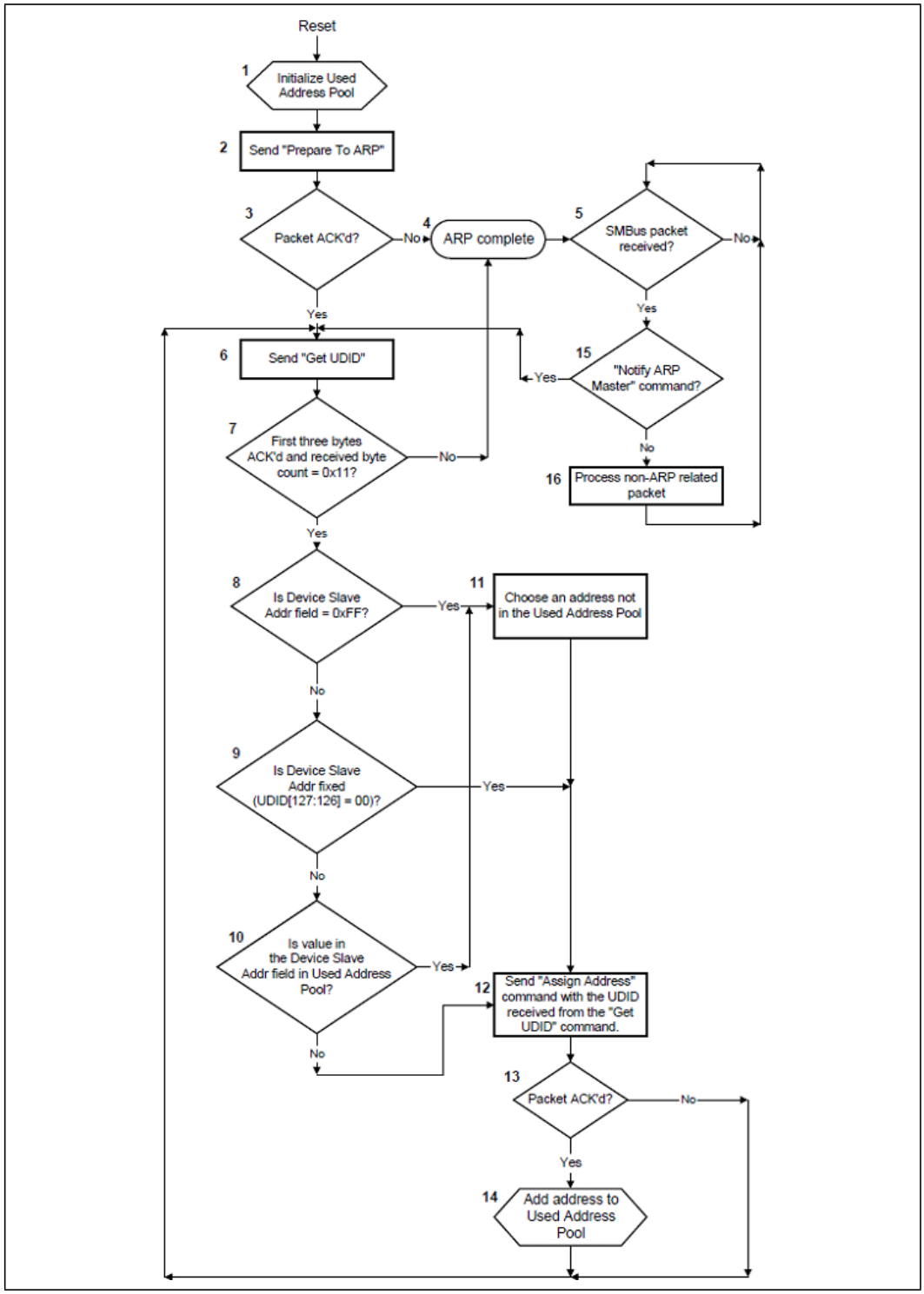
Since the SMT supports multiple UDIDs, during the ARP process it participates in the Get UDID (general) and Assign Address protocols once for each UDID such that each UDID has a slave address which is resolved by the ARP master.

For GPBR and all both Get UDID flavors, the transaction is descriptor-only with no target data written to memory, and the descriptor is managed by the hardware.

Note: The firmware implements a time-out mechanism such that if the SoC issues the Notify ARP Master command and the ARP Master does not respond within a particular time period then the SoC re-issues the Notify ARP Master command. It is implemented to comply with *SMBus 2.0 Specification* for bus timing.

15.4.3.4 ARP Master Behavior

Figure 15-3. ARP Master Behavior Flow Diagram



The ARP Master must always execute the ARP when it enters the working state and anytime it receives an SMBus status change indication (due to device addition or removal). The process begins with the ARP Master issuing the Prepare To ARP command. In all cases the ARP Master must be able to resolve addresses when it receives the Notify ARP Master command.

Since SMBus devices join the system without a corresponding system reset (i.e., a hot-plug event), the ARP Master optionally chooses to issue the Get UDID (general) command at least once every 10 seconds to discover newly added devices that require address resolution but which do not support the Notify ARP Master command. No device whose AR flag is set responds to this command. However, a newly added device enters the system with a power-up reset, which resets its AR flag; it responds to a Get UDID (general) command with its UDID. The host chooses to assign such a newly added device a non-conflicting address or chooses to re-ARP the entire bus.

Until the ARP process is complete the ARP Master must not wait more than tow seconds before issuing a Get UDID (general) command after issuing the previous Get UDID (general) command. This restriction is important to allow another SMBus master to determine when it is safe to do an enumeration of the bus.

With reference to Table 15-10 the ARP Master operates as follows (steps which are shaded are the responsibility of the firmware alone or of the firmware to trigger the hardware, or vice versa).

Table 15-10. ARP Master Operation (Sheet 1 of 2)

Step		Description
0. . . Perform ARP master initialization.		• [FW] See Section 15.4.3.5, "ARP Initialization Flow" on page 302
1.	Upon starting, the ARP Master initializes its Used Address Pool. Initially this consists of the slave addresses of fixed-address SMBus devices known to the ARP Master and reserved addresses (as defined in *SMBus 2.0 Specification*).	• [FW]
2.	Send the Prepare To ARP command.	• [FW]
3.	Check for an acknowledgement for all bytes in the previous packet. If any bytes were not acknowledged then the ARP Master assumes that no ARP-capable devices are present and therefore consider the ARP process complete; proceed to step 4. If all bytes were acknowledged then go to step 6.	• [FW]
4.	The ARP Master found no response to the Prepare To ARP command so it assumes that no ARP-capable devices are present in the system at this time. The ARP Master periodically re-issues the Prepare To ARP command to discover any ARP-capable devices added. Proceed to step 5.	• [FW]
5.	Wait for an SMBus packet. If a packet is received proceed to step 15.	• [FW]
6.	Send the Get UDID command.	• [FW]
7.	Check for an acknowledgement for the first three bytes and verify that the byte count value received is 0x11. If not, then the ARP Master assumes that an ARP-capable device(s) is no longer present and therefore consider the ARP process complete; proceed to step 4. Otherwise proceed to step 8.	• [HW and FW] HW verifies the byte count before sending descriptor to FW.
8.	Check the value of the Device Slave Address received. If 0xFF then proceed to step 11 since this device does not possess a valid slave address. Otherwise proceed to step 9.	• [FW]
9.	Determine if this device has a fixed slave address. If bits 127 and 126 of the UDID are 00b then it has a fixed address, so proceed to step 12. Otherwise proceed to step 10.	• [FW]

Table 15-10. ARP Master Operation (Sheet 2 of 2)

Step		Description
10.	The device possesses a valid slave address. However, the ARP Master must check this address against the Used Address Pool to ensure that no other device has already been assigned the same address. If the received Device Slave Address is found in the Used Address Pool then proceed to step 11. If not, then the device keeps its current slave address but needs acknowledgement from the ARP Master; proceed to step 12.	• [FW]
11.	Select a slave address that is not in the Used Address Pool and proceed to step 12.	• [FW]
12.	Send the Assign Address command with the UDID returned by the device in the Get UDID command packet.	• [FW]
13.	Check for acknowledgement of all bytes in the Assign Address command packet. If any byte was not acknowledged then the ARP Master assumes the device is no longer present; proceed to step 6 to determine if more devices require address resolution. If all bytes were acknowledged then the ARP Master assumes that the device has accepted the address assignment; proceed to step 14.	• [HW and FW] HW monitors slave ACK before sending descriptor to FW.
14.	The device now has a valid slave address. The ARP Master must add this address to the Used Address Pool. Proceed to step 6 to determine if more devices require address resolution.	• [FW]
15.	The ARP Master checks if the received packet was the Notify ARP Master command. If so, then it must execute the ARP to resolve the address for the newly added device(s); proceed to step 6. If not, then proceed to step 16.	• [FW]
16.	The ARP Master received a non-ARP related packet. Process it accordingly and proceed to step 5.	• [FW]

These steps cover the case when the ARP Master has exited a reset state. Since ARP supports hot-plug the Master must be prepared to execute the ARP at any time; step 15 covers the case when a device issues the Notify ARP Master command. If during the ARP process a previously detected device is now not discovered, then the slave address associated with that device is removed from the Used Address Pool.

The diagram does not consider bus time-out mechanisms or retries. These are implemented to comply with SMBus timing requirements.

15.4.3.5 ARP Initialization Flow

The firmware initializes the SoC for ARP protocols before any SMBus traffic. Because the SMT master behavior applies to both master-transmitter (e.g., ARP Host issuing Prepare to ARP) and slave-transmitter (e.g., ARP slave issuing Host Notify), and SMT target behavior applies to both master-receiver (e.g., ARP Host receiving Host Notify) and slave-receiver (e.g., ARP slave receiving Prepare to ARP), many ARP initialization steps are common to Host and slave devices.

The exception is UDID assignment. This pertains only to ARP slaves.

Table 15-11. ARP Initialization Flow (Sheet 1 of 2)

Initialization Flow for an ARP Slave	Initialization Flow for the ARP Host
Configure the SoC bootstrap configuration information programmed fields • Preset UDID fields are loaded from the SoC bootstrap configuration information • Also, the SoC bootstrap configuration information is loaded to configure the unique UDID field (VSID)—the hardware permutes the SoC bootstrap configuration information to program multiple UDIDs as necessary ***Example:*** To support two functions each with two UDIDs, combine three hard-coded LSBs with a sufficient number of SoC bootstrap configuration information bits (n) to distinguish all SoC devices. **Bit Fields:** VSID = [31:32-*n*] [32-*n*-1:3] [2:1] [0] VSID = SoC bootstrap configuration information bits static-bits function UDID Fn0/UDID0 = {SoC bootstrap configuration information bits}{static bits}{00}{0} Fn0/UDID1 = {SoC bootstrap configuration information bits}{static bits}{00}{1} ... Fn1/UDID1 = {SoC bootstrap configuration information bits}{static bits}{01}{1} The SoC bootstrap configuration information bits are shared by all functions and UDIDs. • Preset other fields based on the SoC bootstrap configuration information: GPBRCTRL.GPTRADR, SPGT.SPD, TPOLICY.ADDR0_EN, TPOLICY.ADDR1_EN, and TPOLICY.Host_SMBADDR_EN.	Configure the SoC bootstrap configuration information programmed fields • N/A • N/A • Preset other fields based on the SoC bootstrap configuration information: GPBRCTRL.GPTRADR, SPGT.SPD, TPOLICY.ADDR0_EN, TPOLICY.ADDR1_EN, and TPOLICY.Host_SMBADDR_EN.
Program SMBus address to any fixed-address target • Program TACTRL.ADDR0 if required based on UDID0.DEVCAP field • Program TACTRL.ADDR1 if required based on UDID1.DEVCAP field	
Verify transaction status flags (no invalid state flags) • Verify all 0s in fields TSTS.IP and MSTS.IP • Likewise, verify numerous error-status registers	
Program descriptor base address and buffer entries • Program MDBA and MDS • Program TBBA and TBS • Program SMTICL	
Program head and/or tail pointers • Program MCTRL.FMHP and MSTS.HMTP • Program HTHP.HTBHP and FTTP.FTBTP	

Table 15-11. ARP Initialization Flow (Sheet 2 of 2)

Initialization Flow for an ARP Slave	Initialization Flow for the ARP Host
Enable (unmask) interrupts from the hardware due to certain conditions • Assert/deassert ERRINTMSK, ERRAERMSK, TCTRL.TIE, MCTRL.MEIE	
Configure policies • Program RPOLICY.TBRCLKCNT, RPOLICY.COLRTRY, RPOLICY.TBRBCLK, RPOLICY.RETRY	
Configure GPBR • Set GPBRCTRL.EN • Assert/deassert GPBRCTRL.PECEN • Assert/deassert GPBRCTRL.HWCLRDIS • Program GPBRCTRL.BC • Program GPBRCTRL.CMD • Program GPBRCTRL.GPTRADR	

15.4.4 SMT System Usage Models

The SMT architecture takes into account various usage models as envisioned on the SoC-based platforms. The focus of the architecture is to keep the hardware overhead low for supporting the various usage models and protocols:

- SMBus ARP Mastering or ARP Target
- Embedded Controller (EC) on the SMBus communicating with the SoC

15.4.5 SMT Security Requirements

The SMT has no security requirements.

15.4.6 SMT Timing Modes

The SMT currently supports three different timing modes: standard (up to 100 kHz), fast-mode (up to 400 kHz), and fast-mode plus (up to 1 MHz). The timing requirements for fast-mode and fast-mode plus are found in the I^2C industry specification.

The following section covers design targets for the various timing modes in various operating conditions. The SMT controller does not contain dynamic timing adjustment to account for varying bus loads or pull-up resistor selections, thus SMT is designed and have default settings to meet the specification in the worst case scenarios. This comes at the cost of not having optimal frequency for any given timing mode in nominal conditions.

Note: Devices also stretch the clock low thus reducing the frequency. These timing estimates are assuming the device is not further impacting the timings.

Table 15-12. SMT Timing Mode Maximum Clock Frequency Ranges

Mode	Maximum Clock Frequency Range
Standard	90-100 kHz*
Fast Mode	350-400 kHz*
Fast Mode Plus	750-1000 kHz*

*Timing assumes the platform design is within the specification for maximum rise time allowed requiring appropriate Rpullup value selection for the given capacitive load on the bus.

15.4.7 SMT as Master

The SMT as the initiator provides the hardware for the internal agents inside the SoC to send/receive data across the SMBus. Due to the various usage models that SMT supports, the hardware support exists for initiating reads/writes from/to external devices on SMBus. The SMT has the hardware capability to transport them over SMBus physical and report status back to the firmware.

Note: The firmware must ensure the data in the transmit data buffers is arranged in the order that it wants to send on the bus, i.e., data pointed to by the Tx data pointer are sent first on SMBus, then the next byte, etc.

With respect to the receive buffer, data are written to memory as-is received from the SMBus physical, i.e., the first byte received is placed in the lowest address, then the next byte, etc.

15.4.7.1 Hardware Buffering for Master Support

The hardware implements SMBUS 2.0 support for maximum block transfer of 32B which it uses to queue transactions (both master-transmitter and master-receiver) as a descriptor-based master.

The storage queues are not shared between the master and the target. Both logical sides function independently without dependency on each other.

Note: Although SMT theoretically supports master-initiated write and read cycles of arbitrary length, the practical expectation is that the transaction size does not exceed 32B.

15.4.7.2 Master Descriptor

The master descriptor is a ring buffer of individual descriptors set up by the firmware as shown in Figure 15-4. The descriptor ring buffer is pointed to by the base (MDBA) and the size is indicated by the MD Size register (MDS). See register definitions for details. Individual descriptors have a 64-bit pointer into the data buffer with an expected transmit and/or receive length programmed in the Control field of the descriptors.

The hardware updates the Status WB field in Figure 15-4, Figure 15-5. See Table 15-13 for detailed descriptions. The remaining three Dwords are updated by the firmware.

The firmware always leads and points by the FWmHeadPtr (MCTRL.FMHP). The hardware always points by the HWmTailPtr (MSTS.HMTP).

Figure 15-4. Master Descriptor Ring Buffer

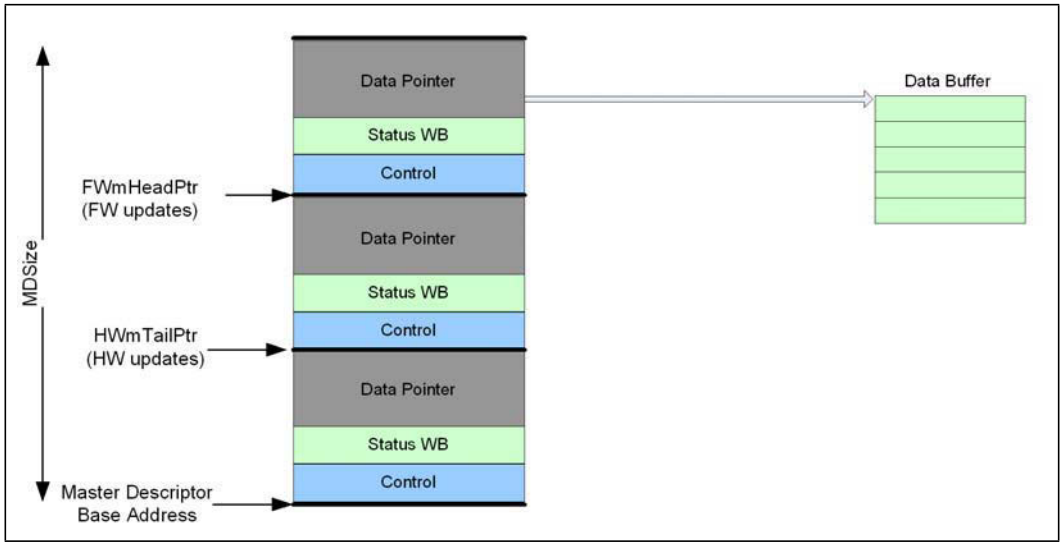


Figure 15-5. Master Descriptor Format

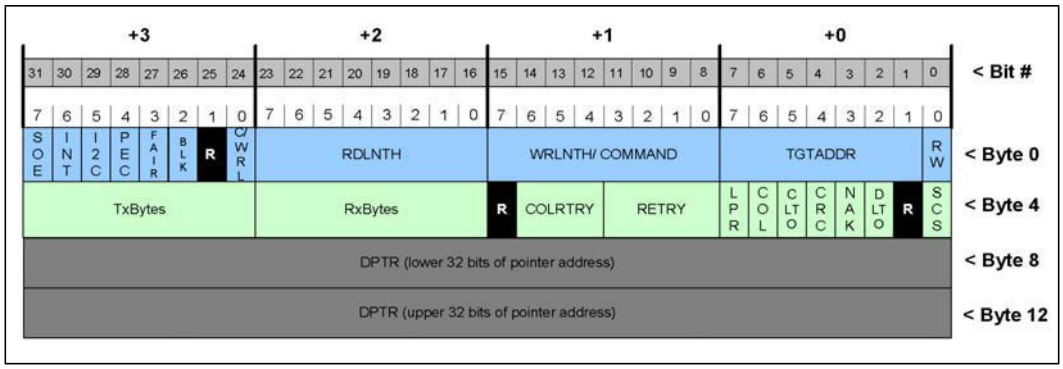


Table 15-13. Master Descriptor Field Descriptions (Sheet 1 of 3)

Dword #	Bit #	Field	Description
0	31	SOE	**Stop On Error**: This bit is set to 1 to direct the hardware that if a descriptor-based master cycle results in an unsuccessful transaction on SMBus, the hardware must clear the start/stop bit (MCTRL.SS) and stop the engine. If this bit is clear, if the hardware encounters an error while sending a transaction on SMBus, it sets the master error status bit (MSTS.MEIS) and continue to process the next descriptor in the queue.
0	30	INT	**Interrupt**: This bit is set to 1 to direct the hardware that it must generate an MSI to the firmware when it has completed the requested transaction successfully. This interrupt is generated only after the status Dword of the descriptor is written back to memory. Interrupts due to unsuccessful transactions on SMBus are not affected by this bit. If this bit is clear and the transaction is successful on SMBus, no interrupt is generated; however, the firmware polls the MSTS.MIS bit. ***Note:*** When using legacy interrupts this bit has no effect; a legacy interrupt is always generated even if not requested by master descriptor (firmware).
0	29	I^2C	I^2C **Enable:** This bit is set to 1 to indicate the hardware must perform the transaction using I^2C protocol.
0	28	PEC	**Packet Error Code**: This bit is set to 1 to indicate the hardware must append CRC (or PEC) as the master of the requested transaction if it is a write or check received CRC (PEC) of the requested transaction if it is a read. PEC is calculated over the entire message (including address and read/write bits) and supplied by the device which puts out the last-data byte of the message.
0	27	FAIR	**Fair**: This bit is set to 1 to indicate if the hardware is able to successfully win arbitration on the bus and master the transaction, it must set its internal fairness flag. This allows a mechanism for fairness on SMBus per *MCTP Specification*.
0	26	BLK	**Block**: Set to 1 by the firmware to indicate the hardware must perform a Block Transaction on the bus. The hardware determines one of three block transactions based on the following fields {BLK, C/WRL and R/W}. 100: Perform SMBus Block Write 111: Perform SMBus Block Read 101: Perform SMBus Block Process Call Others: Reserved
0	25	Reserved	Reserved
0	24	C/WRL	**Command/Write Length**: Set to 1 by the firmware when it has overloaded the Write Length field with the command code of the SMBus transaction.
0	23:16	RDLNTH	**Read Length**: Indicates the number of bytes the hardware receives from the target, and write to the receive data buffer 1-based counting, i.e., the value 0h means 0 bytes of receive data and the value Ah means 10 bytes of receive data. The maximum read length currently supported by the hardware is 240 bytes due to its internal buffer size.
0	15:8	WRLNTH	**Write Length**: Indicates the number of bytes the hardware transmits as master (except the first address byte, byte count, and any subsequent address bytes for reads) 1-based counting, i.e., the value 0h means 0 bytes of transmit data and the value Ah means 10 bytes of transmit data. The maximum write length supported by the hardware is 240 bytes including address due to the size of its internal buffer. The byte count for the SMBus transactions is calculated by the hardware based on the write length field. Example: if WRLNTH = 8, the hardware calculates byte count = 7 and send on the SMBus. The WRLNTH field itself contains command code and 7 bytes of data. When the C/WRL flag is set, this field contains the command code of the SMBus transaction.

Table 15-13. Master Descriptor Field Descriptions (Sheet 2 of 3)

Dword #	Bit #	Field	Description
0	7:1	TGTADDR	**Target Address**: 7-bit address field indicating the target SMBus/I^2C address. ***Note:*** See Section 15.4.7.7, "Write Disabling to DIMM SPD EEPROM Addresses" on page 319 for restrictions on writes to certain addresses.
0	0	RW	**Read/Write**: Set to 1 to indicate a Read Request. Cleared to 0 to indicate a Write Request. Combinations of this bit and the WRLNTH, RDLNTH, and C/WRL fields are decoded by the hardware to distinguish between various types of SMBus cycles and I^2C cycles. ***Note:*** For SMBus Process Call commands, this bit must always be cleared to 0. ***Note:*** See Section 15.4.7.7, "Write Disabling to DIMM SPD EEPROM Addresses" on page 319 for restrictions on writes to certain addresses.
1	31:24	TxBytes	**Transmitted Bytes**: The hardware updates this field to indicate how many bytes transmitted by it were ACKed by the target. This field provides the firmware the ability to reconstruct which particular byte was NACKed by target. This count is 1-based and includes all the bytes sent by the hardware which are ACKed by target including address. Value of 0 means address phase NACKed or collision on address phase.
1	23:16	RXBytes	**Received Bytes**: The hardware indicates how many bytes of received data it is writing to memory (into the data buffer). The count is 1-based, i.e., the value 0h means 0 bytes of data. The hardware limitation is for a 240-bytes data buffer implying that for a maximum SMBus legal block read of 32 bytes with the PEC, it forwards to memory the 240 bytes of data; the byte count received. ***Note:*** PEC is not forwarded to the firmware, since the firmware explicitly enables/disables PEC per master transaction. The hardware returns the PEC check in the CRC field, which the firmware inspects.
1	15	Reserved	Reserved
1	14:12	COLRTRY	**Collision Retry**: The hardware indicates the number of collisions on the last attempt before retiring the descriptor.
1	11:8	RETRY	**Retry Count**: The hardware indicates the count with the number of retries attempted before retiring the descriptor.
1	7	LPR	**Large Packet Received**: The hardware sets this bit to indicate that more data was sent by the target than expected by the firmware and exceeds the allocated receive data space (TRxCTRL.MRxB). The hardware must DMA the data to the buffer in memory in the space allocated, dropping the extra bytes.
1	6	COL	**Collisions**: Set to 1 by the hardware to indicate that failure was due to number of collisions exceeding the collision retry count (RPOLICY.COLRTRY).
1	5	CLTO	**Clock Low Time Out**: Set to 1 by the hardware to indicate unexpected time-out seen on the bus during the course of the request. This bit being set indicates that SMB clock signal was held low by external device for the count as programmed in CNT.
1	4	CRC	**CRC Error:** Set to 1 by the hardware to indicate CRC error[1] on the request. For read requests with PEC, the hardware sets this bit if the PEC received from target does not match PEC calculated by the hardware. For write Requests with PEC, the hardware sets this bit if the target NACKs the PEC byte.
1	3	NAK	**NACK Received:** Set to 1 by the hardware to indicate unexpected NACK asserted by target.
1	2:0	Reserved	Reserved
1	0	SCS	**Success**: Set to 1 by the hardware to indicate that cycle was transferred successfully. 0 indicates that some error was encountered and other bits in the status indicate the error.

Table 15-13. Master Descriptor Field Descriptions (Sheet 3 of 3)

Dword #	Bit #	Field	Description
2	31:0	DPTR	**Data Pointer:** Byte-aligned pointer to the starting location of the data buffer. The hardware reads from memory and transmit exactly as many bytes as is indicated by WRLNTH field. This does not include the address fields of the message (i.e., Start Address/Repeated Start Address) and byte count if any. The hardware is expected to write to memory exactly as many bytes as is indicated by the RDLNTH field (unless an error condition exists and the target fails to provide the expected number of bytes, in which case the appropriate error bits are set).
3	63:32	DPTR	

1. The *SMBus Specification* is vague about the condition of the CRC error when the master is initiating a READ and the target provides the CRC. Irrespective of the fact that the CRC is correct or incorrect, the master must NACK the cycle and assert STOP. The SMT hardware informs the firmware of the incorrectness of the received CRC and leaves it up to the firmware for further action. See the *System Management Bus (SMBus) Specification*, Version 2.0, page 27, paragraph 3.

15.4.7.3 Master Descriptor Usage

Descriptors are used for SMT transactions are initiated by the firmware. The hardware decodes the I^2C, PEC, WRLNTH, and RDLNTH fields to differentiate between types of SMBus transactions (see Section 15.4.7.3.1, "SMBus Protocol Transfers" on page 310) and I^2C transactions (see Section 15.4.7.3.3, "I^2C Protocol Transfers" on page 312).

The master transaction flow is defined in Section 15.4.7.4, "Master Transactions Flow" on page 314.

The master retry flow is defined in Section 15.4.7.6, "Master Retry Flow" on page 318.

15.4.7.3.1 SMBus Protocol Transfers

Table 15-14 illustrates how the hardware uses the control information in the descriptor.

Note:

1. The firmware must ensure that I^2C bit is 0 when initiating SMBus transactions.
2. Target Address and R/W# indication is taken from the descriptor.
3. PEC-enabled write transactions proceed in the same manner except that the hardware appends PEC after the last data phase.
4. PEC-enabled read transactions instruct the hardware to check CRC and report status to the firmware.

Note: See Section 15.4.7.7, "Write Disabling to DIMM SPD EEPROM Addresses" on page 319 for restrictions on writes to certain addresses.

Table 15-14. SMBus Transaction Encodings (Sheet 1 of 2)

SMBus Command	BLK	C/ WRL	WRLNTH	RDLNTH	RW (0=W; 1=R)	DPTR (Points to TX Data)	DPTR (Points to RX Data)
Quick Command	0	0	0	0	Read/ Write	X	X
Send Byte	0	1	Command	0	0	X	X
Receive Byte	0	0	0	1	1	X	Valid and points to a buffer where the received byte is placed.
Write Byte	0	0	2	0	0	Valid and points to 2 bytes (command and 1-data byte).	X
Write Word	0	0	3	0	0	Valid and points to 3 bytes (command and 2-data bytes).	X
Read Byte	0	1	Command	1	1	X	Valid and points to a buffer where the received byte is placed.
Read Word	0	1	Command	2	1	X	Valid and points to a buffer where the received 2 bytes are placed.
Process Call	0	0	3	2	0	Valid and points to 3 bytes (command and 2-data bytes).	Valid and points to a buffer where the received 2 bytes are placed.

Table 15-14. SMBus Transaction Encodings (Sheet 2 of 2)

SMBus Command	BLK	C/ WRL	WRLNTH	RDLNTH	RW (0=W; 1=R)	DPTR (Points to TX Data)	DPTR (Points to RX Data)
Block Write (1 byte)	1	0	2	0	0	Valid and points to 2 bytes (command and 1-data byte); the byte count is calculated by the hardware.	X
Block Write (2 bytes)	1	0	3	0	0	Valid and points to 3 bytes (command and 2-data bytes); the byte count is calculated by the hardware.	X
Block Write (3 bytes or more)	1	0	4 or more[1]	0	0	Valid and points to at least 4 bytes (command and 3-data bytes); the byte count is calculated by the hardware.	X
Block Read (1 byte)	1	1	Command	2	1	X	Valid and points to a buffer where the received 2 bytes are placed (byte count, data byte 1).
Block Read (2 bytes or more)		1	Command	3 or more[2]	1	X	Valid and points to a buffer where at least the received 3 bytes are placed (byte count, data byte 1, etc.).
Block Write-Block Read Process Call[3] (write 1 byte, read N)	1	0	2	N[4]+1	1	Valid and points to 2 bytes (command and 1-data byte); the byte count is calculated by the hardware.	Valid and points to a buffer where at least the received (N+1) bytes are placed (byte count, N data bytes).
Block Write-Block Read Process Call (write 2 bytes, read N)	1	0	3	N+1	1	Valid and points to 3 bytes (command and 2-data bytes); the byte count is calculated by the hardware.	Valid and points to a buffer where at least the received (N+1) bytes are placed (byte count, N data bytes).
Block Write-Block Read Process Call (write >2 bytes, read N)	1	0	4 or more	N+1	1	Valid and points to at least 4 bytes (command and 3-data bytes); the byte count is calculated by the hardware.	Valid and points to a buffer where at least the received (N+1) bytes are placed (byte count, N data bytes).

1. Per the *System Management Bus (SMBus) Specification,* Version 2.0, having a block write of exactly 3 bytes can occur. However, a block write of 3 bytes and a write word have identical signaling on the bus.
2. Per the *SMBus 2.0 Specification*, having a block read of exactly 2 bytes can occur. However, a block read of 1-data byte (address, command, byte count = 1, followed by 1-data byte) and a read word from the target (address, command, DataByte1, DataByte2) have identical signaling on the bus.
3. The sum of the data bytes in the write and read phases must not exceed 32 bytes per the *SMBus 2.0 Specification*.
4. N must be greater than 1.

15.4.7.3.2 PEC-Enabled SMBus Transactions

For writes going out from SMT to the SMBus, the PEC is calculated by the hardware and appended. For reads, the PEC byte is received from the target and verified by the hardware (incorrectness notification is provided to the firmware by the status bits). The PEC byte is not sent to the buffer.

Note: Firmware-specific:

1. If the firmware wants to send a large amount of data (more than 32-data bytes without counting command byte and byte count) to a target using SMBus 2.0 protocol and indicates to the hardware to send data using block write protocol, it is the responsibility of the firmware to break the data into multiple packets and set up individual descriptors for each packet.
2. The firmware honors all SMBus 2.0 rules for transactions of write, read, transcribe of the total data transfer on SMBus being not more than 32 bytes (including BlockWr-BlockRd process call).
3. The firmware sets the PEC bit when initiating a PEC-accompanied read from a target so that the hardware verifies and reports PEC accuracy, or when initiating a PEC-accompanied write transaction so that the hardware automatically appends the PEC byte.
4. The firmware must clear the I^2C bit when initiating SMBus transactions.
5. Because a single pointer is used, for SMBus transactions which both transmit and receive data (i.e., Process Call and Block Write-Block Read Process Call), the received data overwrites the transmitted data. Before beginning the transaction, firmware must copy the transmit data if the transmit data are preserved after the transaction.

15.4.7.3.3 I^2C Protocol Transfers

For I^2C transactions, Table 15-15 illustrates how the hardware uses the control information in the descriptor. The following are the I^2C support limitations:

1. Support is limited to 7-bit addressing mode only.
2. Write-Read Combined format is supported with SMT as master.
3. As a target, the hardware supports only writes initiated by an external master.

Table 15-15. I^2C Commands

I^2C Command	C/WRL	WRLNTH	RDLNTH	RW (0=W; 1=R)	DPTR (points to TX data)	DPTR (points to RX data)
I^2C Writes (MTx-to-SRx)	1	Command (1 byte of write data)	0	0	X	X
I^2C Writes (MTx-to-SRx)	0	2 or more	0	0	Points to transmit buffer containing at least 2 bytes of write data.	X
I^2C Reads (STx-to-MRx)	0	0	1 or more	1	X	Points to the receive buffer where the receive data are placed.

Notes:
1. This table assumes the target address is programmed in the descriptor and is not part of the WRLNTH field.
2. The I^2C bit must be set for all I^2C transactions.
3. PEC is not supported for any I^2C transaction.
4. The BLK bit must be 0 for all I^2C transactions.

Table 15-15. I²C Commands (Continued)

I²C Command	C/WRL	WRLNTH	RDLNTH	RW (0=W; 1=R)	DPTR (points to TX data)	DPTR (points to RX data)
I²C Combined Format (write followed by read)	1	Command (1 byte of write data)	1 or more	0	X	Points to the receive buffer where the receive data are placed.
I²C Combined Format (write followed by read)	0	2 or more	1 or more	0	Points to transmit buffer containing at least 2 bytes of write data	Points to the receive buffer where the receive data are placed.

Notes:
1. This table assumes the target address is programmed in the descriptor and is not part of the WRLNTH field.
2. The I²C bit must be set for all I²C transactions.
3. PEC is not supported for any I²C transaction.
4. The BLK bit must be 0 for all I²C transactions.

15.4.7.4 Master Transactions Flow

All master transactions on the physical SMBus pins use descriptors. The high-level flow typically is:

1. The firmware sets the data structures in memory.
2. The firmware programs the descriptors and the associated hardware.
3. The firmware sets the Start bit to initiate the transactions.
4. The hardware processes the descriptor, first setting the InProgress bit, and completes the transaction.
5. The hardware writes data back to memory (if any).
6. The hardware writes back the status to memory for the processed descriptor.
7. If no more descriptors are processed, the hardware clears the InProgress bit.

15.4.7.4.1 Firmware Assumptions

The assumption is an SMT firmware driver exists that understands the SMT hardware register interface and usage for sending and receiving SMBus messages.

15.4.7.4.2 Initialization

1. The firmware allocates a 64B aligned buffer in memory to be used as the master descriptor ring buffer. Each descriptor is 16B long.
2. The firmware then programs up the MD Base Address (MDBA) register with the lower memory address of the descriptor ring buffer, and MD Size (MDS) register containing the number of descriptors in the ring.
3. The firmware initializes the FWmHeadPtr (MCTRL.FMHP) and HWmTailPtr (MSTS.HMTP) by writing all 0 into it.
4. The firmware also programs interrupts as needed.

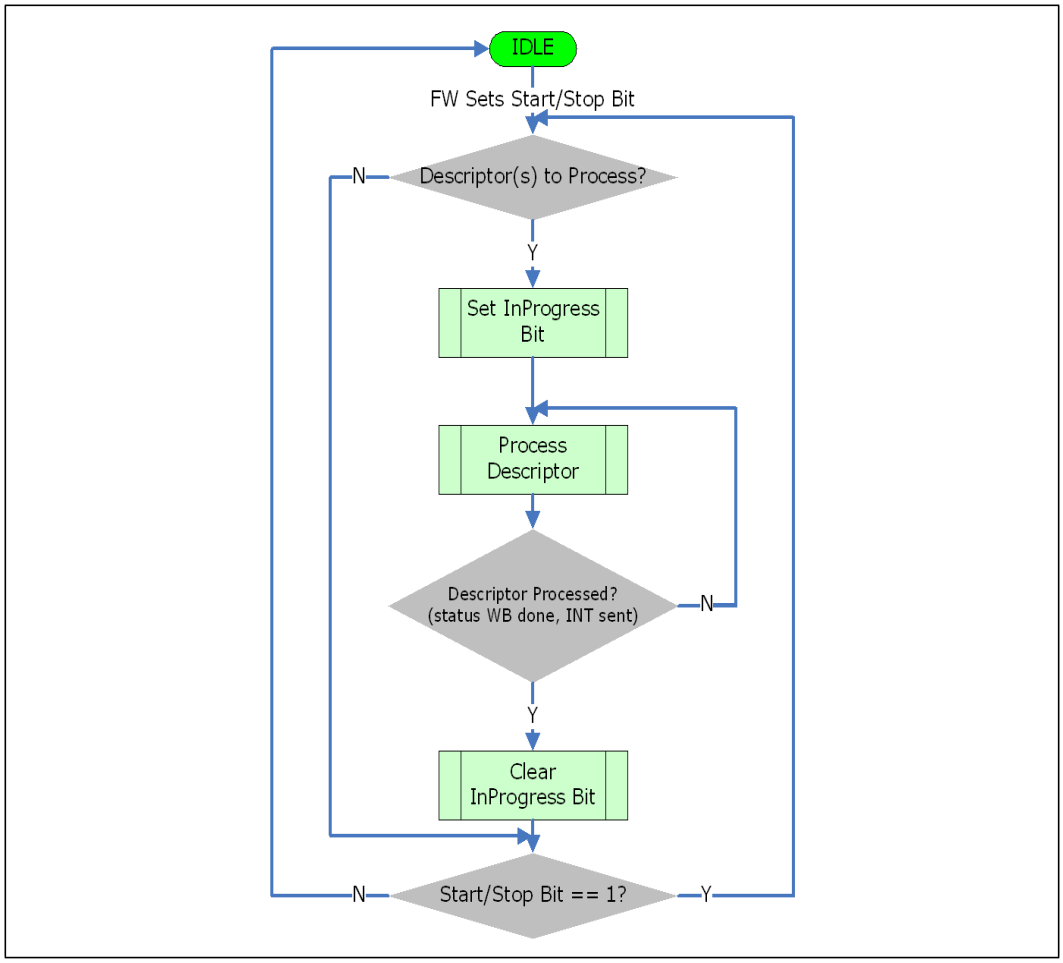
Note: The firmware schedules a master transaction in the descriptor ring buffer if the buffer is not full. The descriptor ring buffer is full if (FWmHeadPtr == HWmTailPtr - 1). The –1 subtraction here needs to account for buffer utilization of N-1 for an N-deep buffer.

15.4.7.4.3 Hardware-Firmware Flow

When the firmware driver has to initiate a transaction on SMBus, and the descriptor ring buffer is not full, the following steps are performed:

1. The firmware programs a 16B descriptor with the attributes of the transaction in memory.
2. The firmware increments the FWmHeadPtr register.
3. The firmware then sets the Start bit (MCTRL.SS) in the hardware.
4. The hardware continuously checks if a descriptor needs to be processed by checking (FWmHeadPtr != HWmTailPtr):
   a. Buffer empty condition: MSTS.HMTP = MCTRL.FMHP.
   b. Buffer full: MCTRL.FMHP = MSTS.HMTP-1 or MCTRL.FMHP-MSTS.HMTP = MDS.
   c. Buffer wrap condition: When MSTS.HMTP = MDS, and MCTRL.FMHP has already wrapped around, i.e., >= 00h, the hardware reads the 16B descriptor, process it and if criteria to increment pointer are met, it wraps to 00h.
5. If a descriptor (see Table 15-6) is available, the hardware first sets the InProgress bit (MSTS.IP), and then reads 16B descriptor from memory by combining (MD Base + HWmTailPtr) from which it:
   a. Decodes the Control Dword for transaction type and other attributes.
   b. Loads the Write Data and Read Data pointers as required.
   c. For writes, the hardware fetches data from the memory pointed to by the DPTR and transmits on wire. For reads, the hardware stores the DPTR pointer so it can Direct Memory Access (DMA) the data to that address when data is provided by the target.
6. Once the transaction is completed, the hardware does a status write back to the Status WB Dword.
7. The hardware then increments HWmTailPtr and send interrupt to the firmware if enabled to do so.
   a. Before issuing the MSI, the hardware denotes the master completion in SMTICL and any error status.
8. The hardware then clears the InProgress bit to indicate it has completed processing of a descriptor.
9. The hardware then checks if the Start bit is set. If so, flow follows from #4, else from #1.

Figure 15-6. Hardware-Firmware Flow Diagram—DMA Mode



15.4.7.5 Clearing of Start Bit

Under certain conditions, the hardware clears the Start bit (MCTRL.SS).

The hardware clears this register for the following fatal error scenarios:

- The transaction is unsuccessful on the final retry (if enabled), including collisions.
- The hardware receives unsuccessful completion on internal IOSF.

The hardware clears this register for the following non-fatal scenario:

- Master descriptor Stop-on-Error (SOE) bit was set and the transaction was unsuccessful.

The firmware clears the Start bit when it kills a master transaction:

1. The firmware clears MCTRL.SS.
2. The firmware sets GCTRL.KILL.
3. The firmware polls MSTS.IP; transaction is killed when bit is detected as cleared.

15.4.7.6 Master Retry Flow

The hardware contains separate timers and counters to auto-retry unsuccessful SMBus cycles that it masters on SMBus.

Two counters for Time Between Retries and Retries Due To Collision are on the bus. Any failed cycle due to collision first exhausts the collision counter before decrementing the retry counter. The Time Between Retries is maintained only between two successive transaction retries and not between two successive retries due to collision.

The high-level flow in pseudo code is:

```
//Master Transaction begin
MASTER_FLOW
// All registers are defined in RPOLICY register
RELOAD RETRY
// Collision retries
COLRTRY
// Time between retries
TBR
While (RETRY >= 0) {
    RELOAD TBR
    RELOAD COLRTRY
    Send Cycle on SMBus
    While (SMBus COLLISION & (COLRTRY > 0)) {
        COLRTRY = COLRTRY - 1
        WAIT for SMBus IDLE
        Send Cycle on SMBus
    }
    If (SMBus ACK) {
    // Successful cycle
    Update Status in descriptor and WB to memory
    Send Interrupt if enabled
    GOTO MASTERFLOW and wait for new cycle
        BREAK
    } else if
    // Unsuccessful cycle
    (((RETRY == 0) & SMBus NACK) || SMBus Timeout) {
    Update Status in descriptor and WB to memory
    Set Master Error and send MSI (if enabled) after updating SMTICL
    Clear DMA Start/Stop Bit
GOTO MASTERFLOW and wait for FW to set Start bit
    BREAK
    } else if (SMBus NACK & (RETRY > 0)){
        RETRY = RETRY - 1
        CNTDOWN TBR
        WAIT for TBR to expire
    }
}
```

15.4.7.7 Write Disabling to DIMM SPD EEPROM Addresses

Although this controller is not intended to participate in DIMM SPD (Serial Presence Detect), an attacker connects its data and clock lines to the segment which includes the SPD EEPROM and mount an attack. To prevent this, a write disable (MCTRL.SPDDIS) is introduced. If SPDDIS is deasserted, writes are not restricted; however, if SPDDIS is set then writes to the address range A0h-AEh are blocked and an error is flagged. Writes to addresses outside the range A0h-AEh are not affected by SPDDIS, and reads are never affected.

Table 15-16. DIMM SPD EEPROM Write-Disable Mechanism

Target Address [7:4]	Target Address [0] (Read/Write Operation)	SPD Disable Bit (MCTRL.SPDDIS)	SMBus Behavior
Ah	0 (write)	0 (enabled)	Allow writes to addresses A0-AEh
Ah	0 (write)	1 (disabled)	Deny writes to addresses A0-AEh and log error (ERRSTS.SPDWE)
Ah	1 (read)	Any	Allow reads to addresses A0-AEh
!= Ah	Any	Any	Allow writes and reads

The SPDDIS is read-write-once, and the BIOS Memory Reference Code (MRC) is expected to set this bit when SPD is complete. Since the SMBus (Host) controller is not intended for SPD, the BIOS must set its SPDDIS at any time in the boot flow.

15.4.8 SMT as Target

The SMT has a fully-functional target interface for other masters on the SMBus intending to communicate with SMT. The usage models for this are:

- SMBus ARP Mastering or ARP Target
- Embedded Controller (EC) on SMBus communicating with the SoC

The hardware aspect of the target interface is highly generalized. Most transactions are treated as raw data which are pushed to the firmware, where the firmware transacts level activity like protocol detection. Acceleration in hardware is limited to interception of ARP Get-UDID and SMBus Block-Read protocols, which require an immediate return of data to an external master, and PEC CRC calculation. The other hardware responsibility is inspection of enough transaction bytes (e.g., target address, command code, UDID) to ascertain the protocol type and to engage the appropriate hardware flow.

15.4.8.1 Hardware Buffering for Target Support

The hardware implements a separate 240-bytes buffer to store the bytes it receives as a target. The hardware only supports SMBUS 2.0 maximum block transfer of 32B.

The hardware also implements a separate 32-byte buffer which provides generic read data to external masters for any reads that they perform. This is a generic usage model in which firmware repeatedly programs the generic read-data buffer register (GPBRDBUF) with data, programs a MMIO offset (GPBRCTRL) with address, command, and byte count of how much data is present in the generic read data buffer, and communicates to the external master the hardware is ready for taking the read (in-band through SMBus). The external master then launches a block read to the address programmed with the command, and the hardware provides the byte count and data bytes associated with it.

15.4.8.2 Target Descriptor

Unlike the master descriptor, the target descriptor is a ring buffer of data and status all existing concurrently. This simplified model results in the hardware writing status of a received cycle immediately followed by the data received in the cycle (if applicable).

See also TRxSTS, which is used for debug.

Figure 15-7. Target Ring Buffer

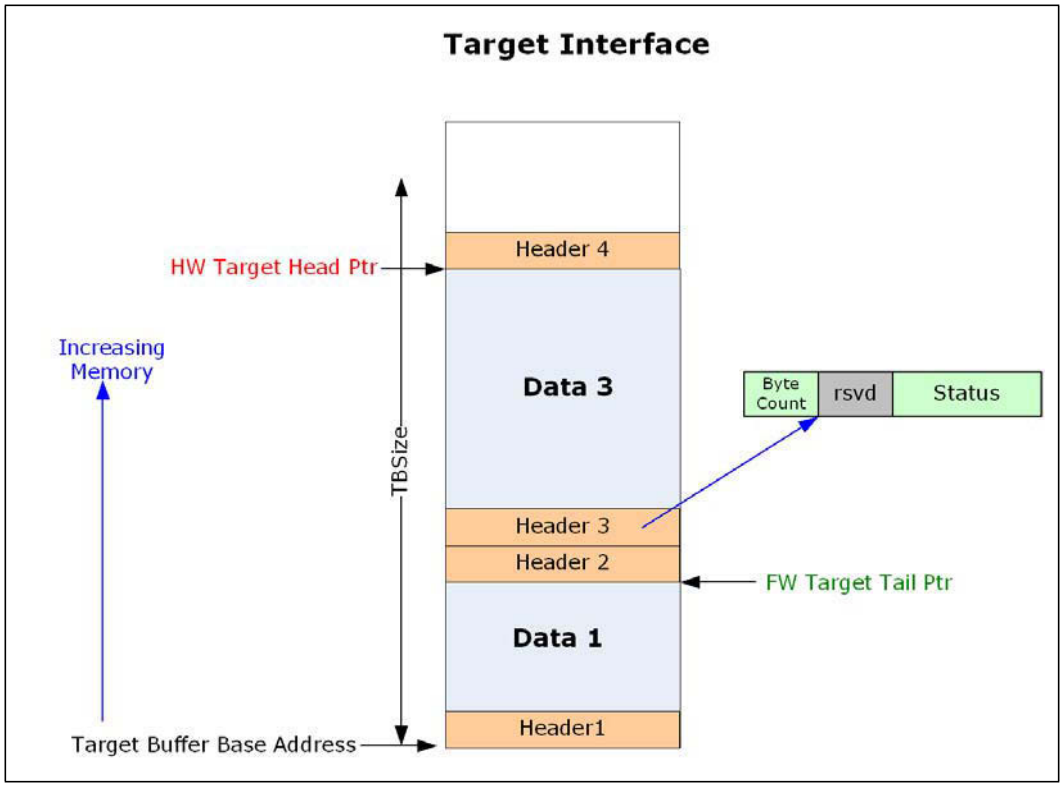


Figure 15-8. Target Header Format

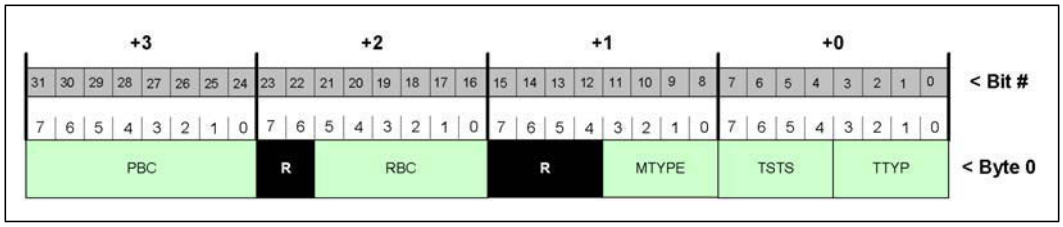


Table 15-17. Target Header Descriptor (Sheet 1 of 2)

Bit #	Field	Description
31:24	PBC	**Payload Byte Count:** The hardware updates this field to indicate how many bytes of payload data follows after the header in the target buffer. For writes coming in, this data include the sum total of all bytes ACKed from start to stop. The hardware does then DMA all those bytes to memory. • For error scenarios where the master drives more data than expected or command code does not match, the hardware does DMA all bytes ACKed on SMBus from Start until the first byte NACKed by the hardware (this byte is not sent to memory), and the total number of bytes DMAed to memory is reflected in this field. A value of all zeros in this field means no payload is following the header.
23:22	Reserved	Reserved
21:16	RBC	**Read Byte Count**: For external master-generated reads, this data include the sum total of bytes provided by the hardware which were successfully transferred over SMBus. For example: for an externally generated block read, the hardware provides byte count, X number of bytes, and PEC (if PEC-enabled transaction). In this case, the RBC field is 1+X+1 (for PEC). ***Note:*** For SMBus Block Reads, the last byte transferred is NACKed by the external master per definition. This is data byte or PEC byte supplied by the hardware. The data bytes are not written to memory, only the byte count indicates how many bytes provided by the hardware were successfully transferred over SMBus. This field is all zeros if the transaction was an externally-generated write. The header is written to memory (if enabled to do so) containing the status of the transaction. For error scenarios in reads, the status register in the header provides details of the error and this field contains incorrect values due to the pipelined nature of the hardware.
15:12	Reserved	Reserved
11:8	MTYPE	**Message Type**: These bits indicate which expected transaction happened on the bus since they are expected in the usage model and some have pre-programmed data provided by the firmware. 0000: SMBus/I^2C transaction 0111: Block Read to General Purpose Read Data Buffer 1000: SMBus ARP Prepare to ARP 1001: SMBus ARP Reset Device (general) 1010: SMBus ARP Get UDID (general) 1011: SMBus ARP Assign Address 1100: SMBus ARP Get UDID (directed) 1101: SMBus ARP Reset Device (directed) 1111: SMBus ARP Notify ARP Master or SMBus Host Notify Others: Reserved ***Note:*** Firmware: The cycle does not progress far enough for the hardware to encode the correct encodings. In this case, the hardware inserts some reserved value.

Table 15-17. Target Header Descriptor (Sheet 2 of 2)

Bit #	Field	Description
7:4	TSTS	**Transaction Status**: This field indicates the overall completion status of the transaction, i.e., success, fail, error condition, etc. Multiple error conditions occur in the same transaction (e.g., the hardware NACKs a byte and instead of external master signalling stop, it continues to toggle clock and eventually holds clock low and causes time-out). The hardware captures the first error condition, i.e., register reads 0011 for the example listed above. Refer to Table 15-19, "Target Header Encodings (TSTS) Per Transaction Type (TTYPE)" on page 324. 0000: Success with no errors. 0001: Speculative PEC error detected (the hardware always presumes PEC-enabled writes received). 0010: Protocol error, i.e., external master violated SMBus protocol. For example this is asserting STOP in the middle of the byte, or stopped toggling clock, or doing an illegal repeated start, collision, etc. 0011: Hardware NACK, i.e., the hardware NACKed one or more bytes as the byte did not match the expected byte of the sequence or exceeded hardware limitations. ***Note:*** This is the hardware overflow and does not comprehend overflow of an individual protocol (e.g., Block Write, received bytes > indicated Byte Count). 0100: External NACK (external master NACKed at least 1 byte supplied by the hardware on externally generated read). 0101: Clock-low time-out 0110: Data-low time-out Others: Reserved
3:0	TTYPE	**Transaction Type**: This field indicates the type of transaction received by the hardware in terms of the various usage models supported: 0000: Default SMBus transaction to C2h 0001: Transaction targeting address (TACTRL.ADDR0) of UDID0 0010: Transaction targeting address (TACTRL.ADDR1) of UDID1 0100: Host Notify transaction targeting 10h 0101: Generic Programmable Block Read to programmed address in GPBRCTRL.GPTRADR Others: Reserved ***Note:*** This definition is consistent with TSTS.TCIP.

Only certain combinations of target descriptor MTYPE and TTYPE are valid. These are summarized in Table 15-18 (shaded cells and all unlisted combinations are invalid). For each valid (MTYPE, TTYPE) pair a reference is included to the detailed hardware flowchart.

Table 15-18. Valid Target Descriptor MTYPE and TTYPE Combinations

		MTYPE			
		ARP-Related **1000** **1001** **1010** **1011** **1100** **1101**	**SMBus/I²C** **0000**	**Notify Host** **1111**	**GPBR** **0111**
TTYPE	0000	Section 15.4.8.5.2, "SMBus ARP Target Flow" on page 332	Invalid	Invalid	Invalid
	0001 0010	Invalid	Section 15.4.8.5.6, "SMBus/I²C Target Flow" on page 337	Invalid	Invalid
	0100	Invalid	Invalid	Section 15.4.8.5.1, "Host Notify Target Flow" on page 332	Invalid
	0101	Invalid	Invalid	Invalid	Section 15.4.8.5.3, "General-Purpose Block Read Flow" on page 334

15.4.8.3 Target Transaction Status

Table 15-19 provides a reference of the encodings of the transaction status (TSTS) nibble in the header WB for all valid target cycles (TTYPE).

Table 15-19. Target Header Encodings (TSTS) Per Transaction Type (TTYPE) (Sheet 1 of 2)

TTYPE Cycle Type	TSTS 0000 Success	TSTS 0001 PEC Error	TSTS 0010 Protocol Error	TSTS 0011 Hardware NACK	TSTS 0100 External NACK	TSTS 0101 Clock Low Time Out	TSTS 0110 Data Low Time Out
0101 GP Block Read	No errors	N/A (the hardware transmits the PEC)	• A collision is detected by the hardware. This happens when the hardware intended to drive NACK and saw an ACK on the bus. • The external master signals stop/restart, etc. in the middle of a byte. • The external master signals stop on a byte boundary before the byte count expired. • The external master continues driving more clocks even after the hardware has provided all the bytes. • The external master drives ACK instead of NACK on the PEC byte sent by the hardware. • Bits [7:1] of the repeated start address do not match the GPBR address. • Bit 0 of the repeated start address is 0, and the policy is to check for 1.	Command code byte received from the external master does not match the expected value.	The external master NACKs at least one of the bytes provided by the hardware, i.e., Byte Count, Data1, Data2, Data3,..., DataN.	The hardware detects the SMBus clock low time-out anywhere in the middle of the transaction that it is actively servicing.	The hardware detects the SMBus data low time-out anywhere in the middle of the transaction that it is actively servicing.

Table 15-19. Target Header Encodings (TSTS) Per Transaction Type (TTYPE) (Sheet 2 of 2)

TTYPE Cycle Type	TSTS 0000 Success	TSTS 0001 PEC Error	TSTS 0010 Protocol Error	TSTS 0011 Hardware NACK	TSTS 0100 External NACK	TSTS 0101 Clock Low Time Out	TSTS 0110 Data Low Time Out
0001 Cycles to ADDR0/ UDID0 or 0010 Cycles to ADDR1/ UDID1 or 0000 Cycles to Default Address	No errors	PEC received from the external master did not match the hardware-calculated PEC. ***Note:*** This is a speculative error only since the firmware confirms it was a PEC-enabled SMBus transaction (see Section 15.4.8.5.6, "SMBus/I^{2}C Target Flow" on page 337).	• A collision is detected by the hardware. This happens when the hardware intended to drive NACK and saw an ACK on the bus. • The external master signals stop/restart, etc. in the middle of a byte. • The external master signals stop on a byte boundary before the byte count expired. • The external master drives more bytes than indicated by the byte count. • The external master generates start instead of stop on a byte boundary as a response to the hardware NACK in the previous data phase.	• The hardware NACKs when the external master drives more data than the limit programmed in TRxCTRL.MRxB • The hardware detects the protocol/ address violation (see Table 15-8 and Table 15-9).	N/A (the hardware does not provide any bytes)	The hardware detects the SMBus clock low time-out anywhere in the middle of the transaction that it is actively servicing.	The hardware detects the SMBus data low time-out anywhere in the middle of the transaction that it is actively servicing.
0100 Host Notify *or* Notify ARP Master	No errors	N/A (no PEC)	• A collision detected by hardware. This happens when the hardware intended to drive NACK and saw an ACK on the bus. • The external master signals stop/restart, etc. in the middle of a byte. • The external master signals stop on a byte boundary before it sends 3-data bytes. • The external master drives more than 3-data bytes after the address phase.	N/A (the hardware does not check any bytes)	N/A (the hardware does not provide any bytes)	The hardware detects the SMBus clock low time-out anywhere in the middle of the transaction that it is actively servicing.	The hardware detects the SMBus data low time-out anywhere in the middle of the transaction that it is actively servicing.

Note: All reference to TRxCTRL.MRxb registers must take into account the description of MRxB, i.e., the value is not directly matched since it takes into account some arithmetic based on I^2C/MCTP protocol.

Note: For Time-out Detection as a target, the hardware must be actively servicing the transaction when the time-out occurs. This means the hardware must have at least ACKed the address phase on SMBus and must be committed to the transaction based on the settings of the policies in SUSCHKB register.

15.4.8.4 Target Memory Buffer Hardware-Firmware Flow

As described in the target descriptor section, a ring buffer is maintained by the firmware where the hardware sends the notifications to the firmware. The various notifications are:

1. External master initiates a read: notification of success/failure is sent to the firmware in header. Read data is returned by the hardware (data is pre-programmed by the firmware).
2. External master initiates a write: notification of success/failure is sent to the firmware in header. Write data received by the hardware is sent as payload to memory.

Note:

1. Target buffer is empty when HWtHeadPtr = FWtTailPtr (HTHP = FTTP).
2. Target buffer is full when (HTHP = FTTP - 4B) or (HTHP - FTTP = TBS).
3. Buffer wrap for HTHP is when HTHP = TBS. When this exists, the hardware writes Dword to memory and then roll over to 0x0000 unless buffer full condition exists, i.e., FTTP = 0x0000.
4. The firmware must never increment the FWtTailPtr to a value greater than the HWtHeadPtr.
5. If the target ring buffer is N-bytes deep, only N-1 bytes are utilized since the hardware/firmware do not implement a wrap bit.

15.4.8.4.1 Initialization

1. The firmware allocates a buffer in the firmware memory to be used as the target ring buffer as one contiguous space.
2. The firmware then programs up the Target Buffer Base Address (TBBA) register with a 64B aligned memory address of the ring buffer.
3. The firmware assigns the Target Buffer Size (TBS) register with the actual size of the ring buffer with the maximum limit of 64 KB.
4. The hardware reset initializes the HWtHeadPtr and FWtTailPtr to 0.
5. The firmware enables all target addresses as needed, and program the register-based read data as needed.
6. The firmware also programs interrupts as needed.
7. Finally the firmware sets the TPOLICY.TGTEN bit to enable the target logic.

15.4.8.4.2 Hardware-Firmware Flow

Once the firmware has completed initialization, the hardware target logic continues to idle until an external SMBus master sends a transaction.

External Master Initiating Writes

1. Once the hardware receives a write transaction targeting one of its slave addresses, the hardware responds to the address and command phases.
2. Each byte ACKed is eventually DMAed to the memory starting at location Base + HWtHeadPtr + 4B as Dword writes.
3. Once the transaction is completed, the hardware writes the final bytes to memory (PBC field in the header indicates to the firmware how many true bytes are in the payload).
4. Upon completion of the transaction, the hardware writes the header Dword of that transaction into the location pointed to by HWtHeadPtr. This contains all the status information for the transaction. (See Figure 15-9, “High-Level Target Flow” on page 330.)
5. The hardware then updates the HWtHeadPtr to the next free Dword location in the target memory.
6. The hardware then writes interrupt information to the Dwords as pointed to by SMTICL. (See Table 15-20, “Target Transaction Behavior Due to SUSCHKB.IRWST” on page 331.)
   a. The hardware writes the current value of the HWtHeadPtr to TRGT.HTHP and sets TRGT.VALID.
   b. Also, if the transaction terminated abnormally the error condition is written (e.g., to ERR.TRBAF, ERR.TRBF, ERR.CKLTO) and ERR.VALID is set to indicate an error was present.
   c. Finally, if enabled, the hardware then sends MSI to the firmware.

External Master Initiating Reads:

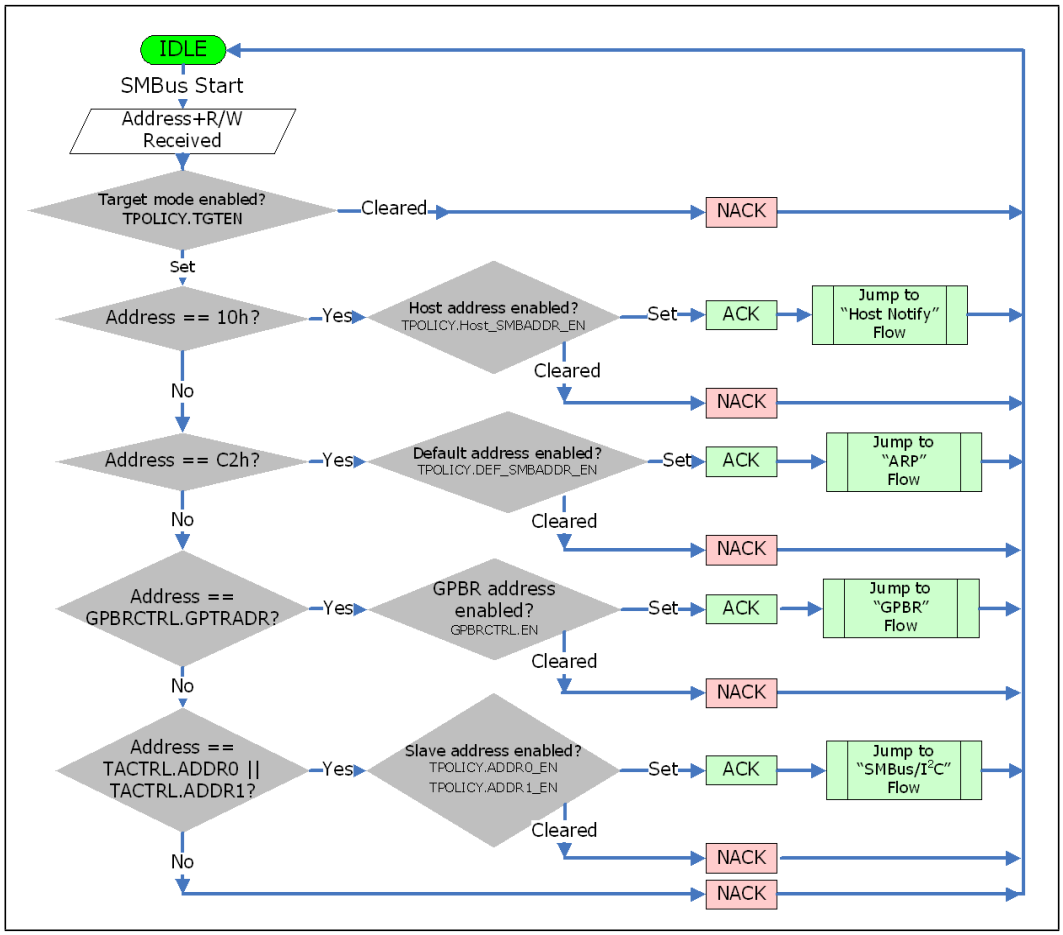
1. Once the hardware receives a read transaction targeting one of its slave addresses, the hardware checks the address, command, and any other bytes sent by the external master and ACK/NACK appropriately.
2. If all the bytes sent by the external initiator are ACKed by the hardware, the hardware provides read data to initiator. (See Figure 15-9, “High-Level Target Flow” on page 330.)
   a. If initiator takes all bytes as expected and terminates transaction normally, depending on the state of TCTRL.SCHWBP bit, the hardware creates (or does not create) a header Dword and write to memory at location pointed by HWtHeadPtr, with BC field updated.
   b. If error conditions are read, the hardware updates the error status registers in the status Dword, create the header with the appropriate BC field, and write to memory at location pointed by HWtHeadPtr.
3. The hardware then updates the HWtHeadPtr to the next free Dword location in the target memory.
4. The hardware then writes interrupt information to the Dwords as pointed to by SMTICL. (See Table 15-20, “Target Transaction Behavior Due to SUSCHKB.IRWST” on page 331.)
   a. The hardware writes the current value of HWtHeadPtr to TRGT.HTHP and sets TRGT.VALID.
   b. Also, if the transaction terminated abnormally the error condition is written (e.g., to ERR.TRBAF, ERR.TRBF, ERR.CKLTO) and ERR.VALID is set to indicate an error was present.
   c. Finally, if enabled, the hardware then sends MSI to the firmware.

15.4.8.5 Target Flow

Target flow for the hardware depends on the target address seen on the SMBus. The hardware has deterministic behavior based on the address. A high-level flowchart is shown in Figure 15-9. It does not capture all possibilities of errors, but aims to highlight the sequence of events. Depending on the received address, detailed flowcharts follow in later sections for four protocol categories.

The hardware address matching after an SMBus start condition is subject to SUSCHKB.IRWST; if asserted it causes the hardware to ignore the R/W# bit during matching. This behavior is not shown in the flowchart. It does have implications for how Quick Command, Receive Byte, and I^2C Read are handled.

Figure 15-9. High-Level Target Flow



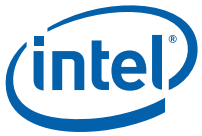


Table 15-20. Target Transaction Behavior Due to SUSCHKB.IRWST

	Quick Command		Receive Byte	I^2C Read
IRWST	**R/W# Bit = Read**	**R/W# Bit = Write#**	**R/W# Bit = Read**	**R/W# Bit = Read**
0	Transaction is ACKed and then dropped. No descriptor is sent. The hardware incorrectly invalidates a supported command option.	Transaction is ACKed and a descriptor is sent to ring buffer for the firmware.	Transaction is ACKed and then dropped. No descriptor is sent. The hardware invalidates this unsupported command.	Transaction is ACKed and then dropped. No descriptor is sent. The hardware invalidates this unsupported command.
1	Transaction is ACKed and a descriptor is sent to ring buffer for the firmware.	Transaction is ACKed and a descriptor is sent to ring buffer for the firmware.	Transaction is ACKed and a descriptor is sent to ring buffer for the firmware. The command is aliased and treated by the hardware as Send Byte. The firmware must invalidate (drop) this unsupported command by detecting the R/W# bit is set in the descriptor.	Transaction is ACKed and a descriptor is sent to ring buffer for the firmware. The command is aliased and treated by the hardware as I^2C Write. The firmware must invalidate (drop) this unsupported command by detecting the R/W# bit is set in the descriptor.

Note: Based on the interaction between SUSCHKB.IRWST and certain supported and unsupported commands, the following recommendations are made:

- If Quick Command is unused (or if its usage is restricted to R/W# bit cleared), then IRWST is cleared. The hardware invalidates the unsupported commands Receive Byte and I^2C Read.
- Otherwise, IRWST is set, and the firmware invalidates (drops) both of these unsupported commands.

15.4.8.5.1 Host Notify Target Flow

TTYPE = 0100 and **MTYPE = 1111**

The host notify flow comprehends both ARP host notification and ordinary SMBus host notify protocols since they differ only in the content of the second received byte. If the firmware has declared this address to be busy (TPOLICY.HOSTBSY) then the hardware will ACK the address byte but NACK all subsequent bytes.

15.4.8.5.2 SMBus ARP Target Flow

TTYPE = 0000 and **MTYPE = 1000, 1001, 1010, 1011, 1100, 1101**

The ARP target flow comprehends all ARP protocols, including general and directed flavors. The priority shown in the flowchart for decoding the command byte is illustrative only. All ARP-related transactions require PEC processing: for Get UDID the hardware must calculate and transmit the PEC; in all other protocols the hardware must calculate the PEC and compare that against the received PEC byte.

If the firmware has declared this address to be busy (TPOLICY.C2_BSY) then the hardware will ACK the address byte but NACK all subsequent bytes.

The hardware address matching after an SMBus repeated-start condition is subject to SUSCHKB.IRWRST; if asserted it causes the hardware to ignore the R/W# bit during matching. This behavior is not shown in the flowchart.

Figure 15-10.Host Notify Target Flow

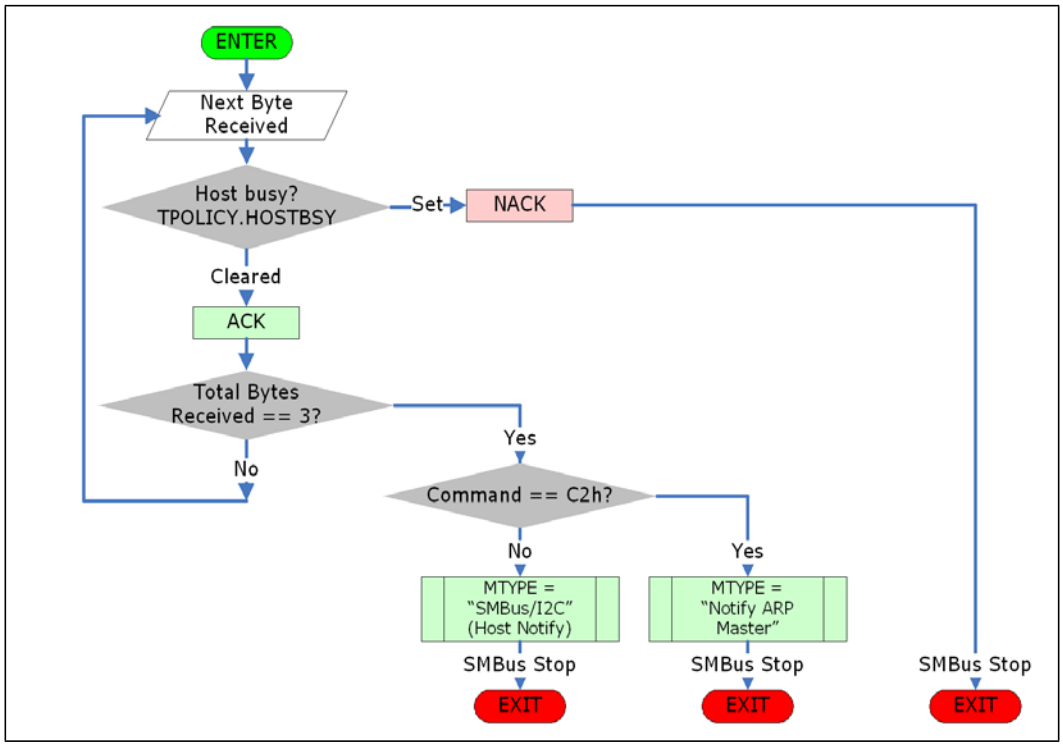
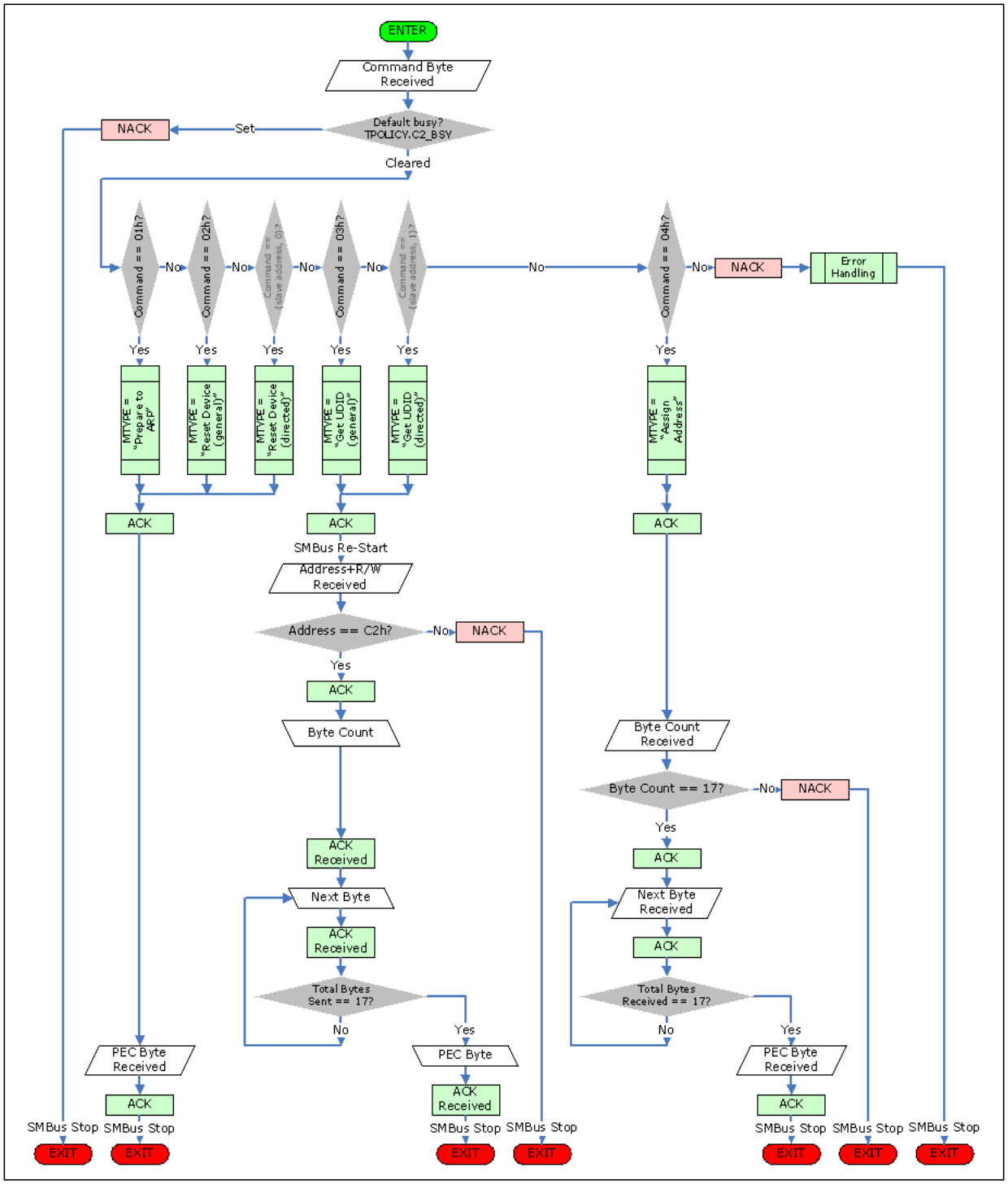


Figure 15-11.SMBus ARP Target Flow



15.4.8.5.3 General-Purpose Block Read Flow

TTYPE = 0101 and **MTYPE = 0111**

SMT provides a mechanism for an external master to read up to 32B of data from SMT.

The rules are:

- External master cannot asynchronously launch a read request.
- Address to read from is programmed by the firmware (GPBRCTRL.GPTRADR) and this address must be unique from all other target addresses that SMT supports.
- External master must read data only as defined by an SMBus 2.0 Block Read command.
- The firmware programs the data into the Data Buffer register (GPBRDBUF) and enables the hardware to support a read command from the external master (GPBRCTRL.EN).
- The firmware sends a message to external master requesting it to read the data.

The sequence of events is indicated by the flowchart shown in Figure 15-12.

The hardware matches the received command code against a firmware-programmed expected command (GPBRCTRL.CMD). A mismatch causes the transaction to be terminated with a NACK. This matching is required since GPBR is handled entirely in the hardware; the firmware does not participate in validating the address and command.

An appended PEC is optional and is enabled by the firmware (GPBRCTRL.PECEN). Its usage is determined by high-level policy negotiation between the SoC and the external master.

Nominally, the hardware clears its enable bit immediately after ACKing the command byte. This enables a handshake between the hardware and the firmware such that the hardware transmits read data only after being programmed and its trigger set by the firmware. Alternately, the hardware is enabled (GPBRCTRL.HWCLRDIS) to continually respond to read requests from an external master; however, in this mode, determinism is not guaranteed between the firmware programming to and master reads of the data registers.

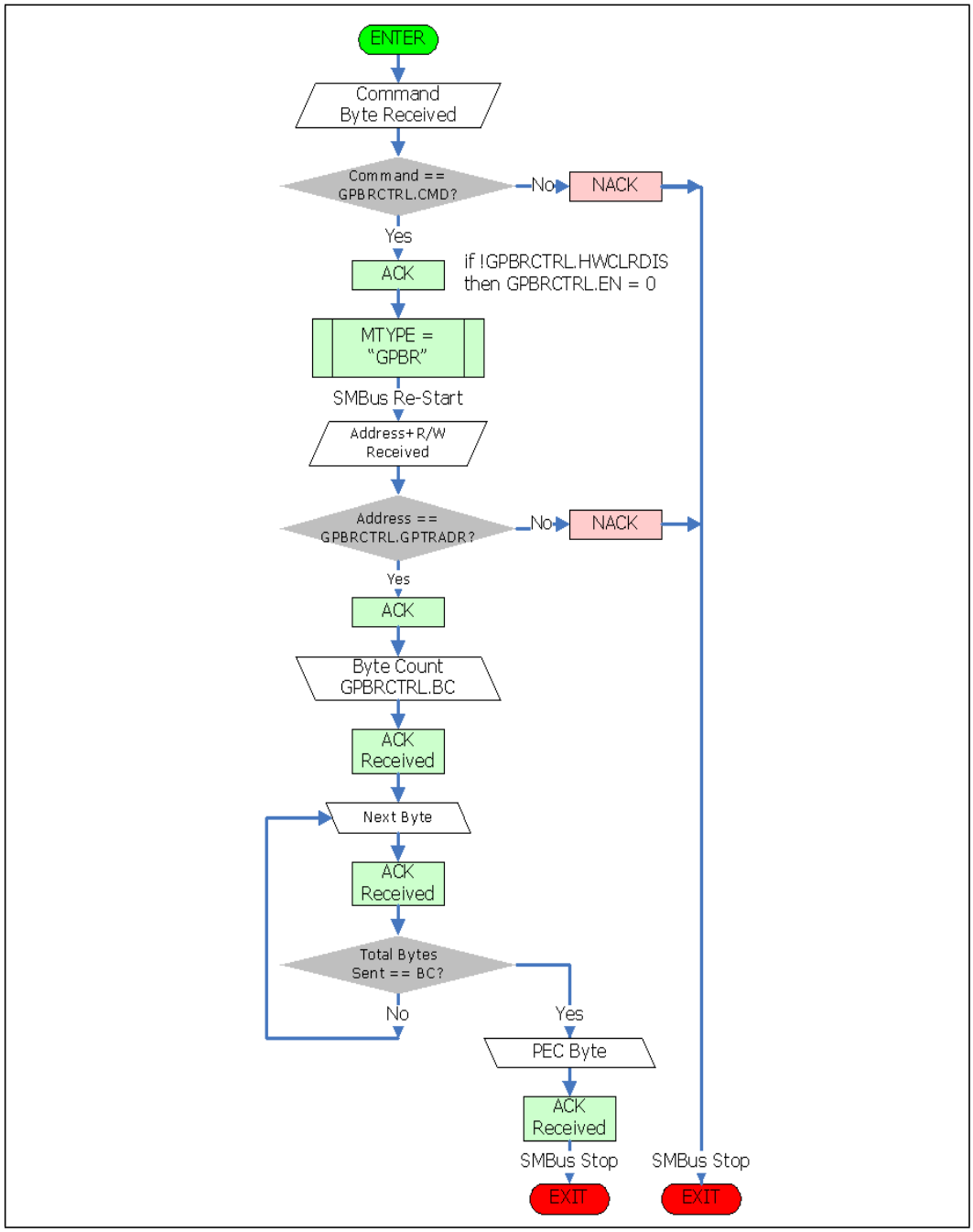
The hardware address matching after an SMBus repeated-start condition is subject to SUSCHKB.IRWRST; if asserted it causes the hardware to ignore the R/W# bit during matching. This behavior is not shown in the flowchart.

The generic programmable read data buffer register (GPBRDBUF) loads data into the generic programmable read data buffer. The buffer provides a mechanism of supporting generic programmable reads (GPBR) by an external SMBus master under the firmware initiated flow.

The data buffer is filled by the firmware initiating DW writes to this register. A hidden pointer always points to the next available buffer location—reading and writing to GPBRDBUF cause the pointer to increment automatically.

Note: Because the pointer auto-increments after each read and write, a read of GPBRDBUF does NOT return the most recently written data. Normal usage requires the buffer contents to be cleared before filling it with a new data payload; therefore a read (of the next empty location) returns 0s. (See also GPBRCTRL.BUFRST and GPBRCTRL.PTRRST.)

Figure 15-12.General Purpose Block Read with PEC Target Flow



15.4.8.5.4 Normal Usage

Before programming new data into the GPBR buffer, the firmware sets both GPBRCTRL.BUFRST (which resets the buffer contents to 0) and GPBRCTRL.PTRRST (which resets the buffer pointer position). After the hardware has completed the reset, it clears both bits.

15.4.8.5.5 Debug Usage

To enable the firmware to readback (verify) the contents of the GPBR buffer, the firmware sets only GPBRCTRL.PTRRST, which resets the buffer pointer position but leave the buffer contents intact. Consecutive reads then return the previously written buffer data.

The hardware provides read data payload starting from bits [7:0] of Dword0 of the buffer, then bits [15:8] of Dword0, and so on. All bytes must be programmed in the sequence to be sent out without gaps. The hardware counts the number of bytes indicated in the GPBRCTRL.BC register and start sending them out sequentially starting at byte 0 of Dword 0 until the byte count is decremented to 0.

15.4.8.5.6 SMBus/I^{2}C Target Flow

TTYPE = 0001, 0010 and **MTYPE = 0000**

SMT supports being a target of certain SMBus and I^{2}C transactions (see Figure 15-13, "SMBus/I^{2}C Target Flow" on page 338), and it acknowledges transactions directed to either of its two target addresses (TACTRL.ADDR0 or TACTRL.ADDR1). The basic flow is depicted in Figure 15-13. In general the target hardware interface is agnostic of command and PEC bytes. The firmware handles these bytes.

Note: It is assumed that block reads to SMT as a target are directed only to the dedicated GPBR address (GPBRCTRL.GPTRADR) and are not considered in this flow.

Write cycles received by the SMT target interface nominally terminate when the external master produces a Stop condition. Premature hardware NACK occurs if the interface is busy or overflows. The SMBus Quick Command is comprised only of the address byte, with the data encoded entirely within the R/W# bit; all other SMBus and I^{2}C transactions have additional data bytes.

The optional PEC byte is not shown explicitly because the firmware determines the actual presence of PEC. This provides for PEC to be enabled on a per-function, per-address, per-protocol, or even per-transaction basis. The hardware always performs a **speculative** internal PEC calculation, which it then compares to the final received byte. If the received and calculated bytes match, this is a strong indication that the transaction was PEC-enabled, but the firmware has the final authority. Conversely, a mismatch is not proof of a transmission error—it may have been a PEC-less transaction. In this case, the firmware ignores the PEC status flag in the target descriptor TSTS field, which resulted from the hardware invalid speculative PEC comparison.

Since some protocols have identical transmission templates it is also the responsibility of the firmware to prevent or resolve apparent aliasing. (e.g., aliasing of Send Byte with PEC and Write Byte; aliasing of Write Byte with PEC and Write Word; aliasing of SMBus Write Byte and I^{2}C MTx-to-SRx.) It is assumed this is accomplished through negotiation between the firmware and the external master. For instance, the command byte differentiates protocols or certain protocol combinations are excluded.

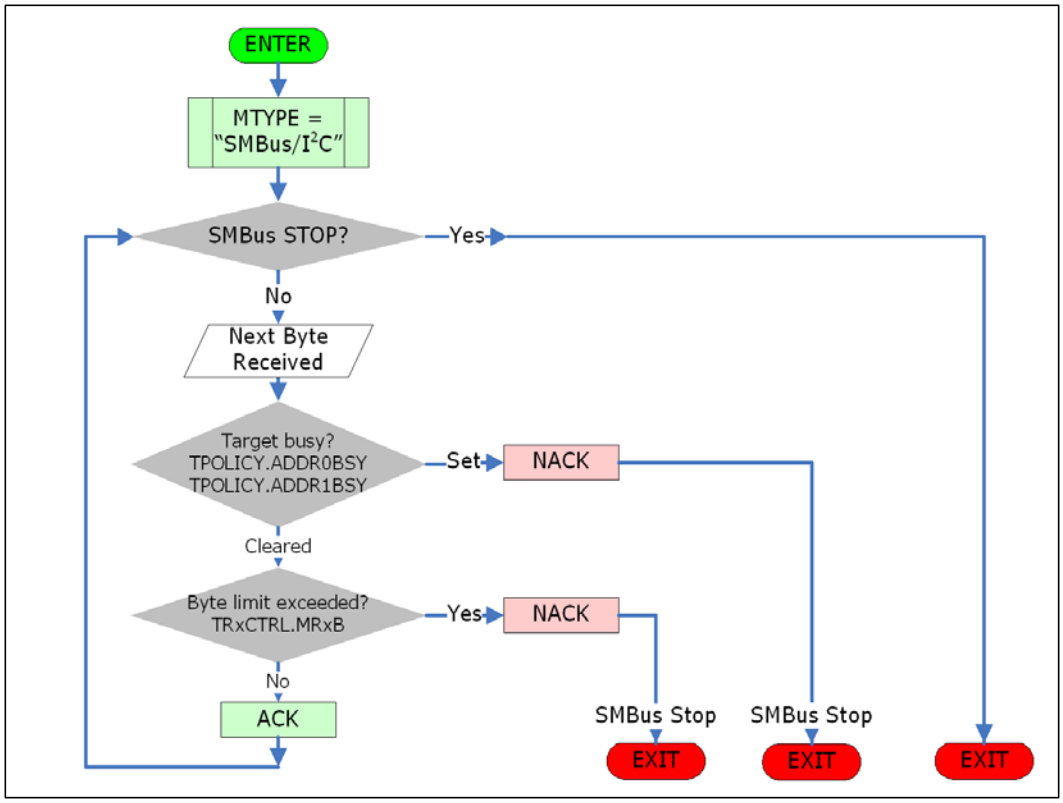
A special case of aliasing occurs for SMBus Block Write. Since the hardware ignores the command byte, it cannot distinguish Block Write Byte Count from an ordinary data byte. Therefore, the hardware captures and pushes all bytes to the firmware for processing, and the hardware is not required to verify the received byte count. In target mode, the firmware programs a ceiling (TRxCTRL.MRxB) on the write length such that if a transaction overflows the ceiling then the hardware must NACK any unexpected data bytes. This ceiling must not exceed the buffer size. The hardware NACK flag in the target descriptor TSTS field is set upon a hardware (buffer) overflow, but this flag does not comprehend overflow of an individual protocol (e.g., a Block Write where received bytes > indicated Byte Count).

Note: If a target address receives both I^{2}C and SMBus protocols the maximum permitted write length (TRxCTRL.MRxB) must accommodate both. For example, if a target receives SMBus Block Write with PEC the write length must be greater than or equal to 36 bytes (= address + command + byte count + N data bytes + PEC, where N <= 32).

If the firmware has declared a matched address to be busy (TPOLICY.ADDR0BSY or TPOLICY.ADDR1BSY) then the hardware will ACK the address byte but NACK all subsequent bytes.

The hardware address matching after any SMBus repeated-start condition is subject to SUSCHKB.IRWRST; if asserted it causes the hardware to ignore the R/W# bit during matching. This behavior is not shown in the flowchart.

Figure 15-13.SMBus/I²C Target Flow



15.4.9 Dynamic SMT Policy Update

The hardware provides a mechanism for the firmware to change the policies of the hardware and other registers by the firmware without requiring a reset to the system.

15.4.9.1 Master Policy

For updates to master side of the registers, the firmware already has the control by stopping the hardware DMA engine and halting new master transactions. This ensures that the hardware master side is completely disabled and the firmware chooses to update the policy and registers.

15.4.9.2 Target Policy

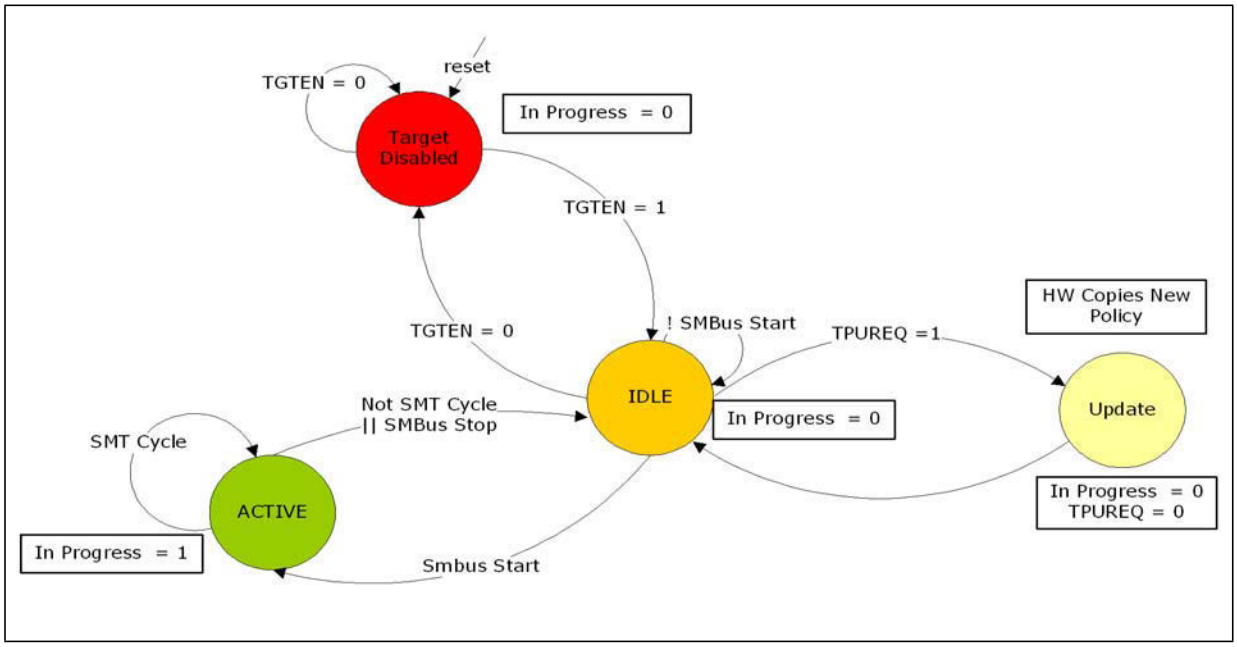
Updates to target policy require a flow, since the hardware asynchronously receives a target cycle.

The hardware maintains current target policy internally without direct visibility to the firmware. The firmware sees current policy by reading TPOLICY register. When TPOLICY.PTUREQ=0, it means the current policy settings are effective in the hardware (assuming the firmware follows flow and did not write to any other register without first setting the PTUREQ bit). To change policy, the firmware first disables the target completely or selectively disables addresses so while the firmware is extensively updating registers, the hardware NACKs appropriately in the address or command phase as desired. After that is done, the firmware must again update policy following same flow to have the final desired policy in effect.

1. The firmware sets the intermediate policy desired by programming the TPOLICY register with TPOLICY.PTUREQ also set.
2. When the hardware reaches the SMBus idle state, it checks if PTUREQ is:
   a. If set, the hardware captures current setting of policy in PTUREQ and load into its internal registers. It also clears the PTUREQ bit once the policy is captured.
   b. If clear, the hardware continues to be in idle until another SMBus start is seen.
3. The hardware continues to act on intermediate policy resulting in NACK on address phase or on command phase of incoming cycles (programmed by the firmware).
4. The firmware polls or reads the PTUREQ register to make sure it is clear before reprogramming the registers, etc.
5. Once the firmware is ready to ungate the hardware, the firmware again follows step #1 in which it sets the new policy.
6. Flow again follows steps 2-4 before the final firmware policy is set and active in the hardware.

A conceptual flow is shown in Figure 15-14.

Figure 15-14.Target Dynamic Policy Update



15.5 Interrupts

SMT hardware has the following causes of interrupts (if enabled).

1. Master Interrupt - The hardware generates master interrupts for transactions ending successfully or in failure.
2. Target Interrupt - Generated whenever the hardware writes data and/or the header to memory.
3. Error Interrupt - Generated whenever the hardware detects a bus error or a resources error.

Table 15-21. Summary of SMT Interrupt Enables and Sources

Interrupt Enable/Mask	Interrupt Source	Governed By	Asserted Upon
Master Interrupts			
Master Descriptor INT field	MSTS.MIS		Successful completion of master cycles
MSTS.MEIE	MSTS.MEIS		Unsuccessful completion of master cycles
Target Interrupts			
TCTRL.TIE	TSTS.TIS	TCTRL.SCHWBP TCTRL.UCHWBP TCTRL.URxTWP	Successful completion of target cycles Unsuccessful completion of target cycles Unsuccessful completion of write to target
Error Interrupts			
ERRINTMSK.CKLTO	ERRSTS.CKLTO		SMBus clock-low time-out
ERRINTMSK.TRBF	ERRSTS.TRBF		Target ring buffer is full
ERRINTMSK.TRBAF	ERRSTS.TRBAF		Target ring buffer is almost full
ERRINTMSK.IHIE	ERRSTS.IHIE		Read completion with non-successful status
ERRINTMSK.IMAE	ERRSTS.IMAE		Request is made when PCICMD.BME is clear
ERRINTMSK.ITE	ERRSTS.ITE		Error in IPI transmit transaction
ERRINTMSK.IRDPE	ERRSTS.IRDPE		Data parity error in IPI receive transaction
ERRINTMSK.IRE	ERRSTS.IRE		Error in IPI receive transaction
ERRINTMSK.SPDWE	ERRSTS.SPDWE	MCTRL.SPDDIS	Write is attempted to SPD address range
ERRINTMSK.CPE	ERRSTS.CPE		CSR parity error

Note: First-error cause information is logged in ERRINFO. Also, error interrupts also support masking of escalation to Advanced Error Reporting (AER); see ERRAERMSK.

Note: ERRINTMSK.IMAE: this bit is assumed to be 1 when MSICTL.MSIE is set. IMAE occurs only if PCICMD.BME is clear and MSI requires BME to be set.

15.5.1 Master Interrupts

Two causes for a master interrupt are:

1. Successful - The Interrupt bit in the descriptor is set to indicate the hardware must generate an interrupt on successful completion of the descriptor. An Interrupt Status Register bit is defined in the MMIO space (MSTS.MIS), which is set every time the hardware retires a descriptor. The enable for this interrupt is defined within each individual descriptor.
2. Failure - The descriptor is not successfully completed due to an SMBus error. This interrupt has a Cause Enable (MCTRL.MEIE) and a Cause Status bit (MSTS.MEIS) in the device MMIO space.

Upon completion of descriptor processing, either MSTS.MIS (successful completion) or MSTS.MEIS (unsuccessful completion) must be set by the hardware. This distinction is required so that if the firmware disables MSI (i.e., chooses to operate in polling mode) by polling both flags, it determines when the descriptor has completed and its completion status.

The master interrupts are sent for descriptor-based transactions. Therefore, they are serialized and ordered with respect to the descriptor writeback and specifically tied to the descriptor just processed due to operand requirement.

1. Run descriptor-based master transaction on SMBus.
2. Perform Descriptor Status WB to memory. (See Status WB field in Figure 15-4, “Master Descriptor Ring Buffer” on page 306, Figure 15-5, “Master Descriptor Format” on page 306, and Table 15-13, “Master Descriptor Field Descriptions” on page 307.)
3. Write 1 to the appropriate Cause Status bit (MCTRL.MEIS or MSTS.MIS).
4. If that appropriate interrupt is enabled (locally and globally), MSI is sent and the cause status is auto-cleared, else MSI is not sent and the cause remains set and the firmware is expected to poll the response.

Note: MSTS.MEIS itself merely indicates an unsuccessful completion. The descriptor Status WB field contains full details of transaction and error conditions.

Note: Firmware implementation:

1. If the cause is set for a previous descriptor-based transaction and the firmware enables the global (and local) interrupt enable, the hardware does not send an interrupt for a previously set cause.
2. It is the responsibility of the firmware to ensure that if the cause is set, the previous transactions are accounted for before enabling the MSI (globally and locally)

15.5.2 Target Interrupts

Since the target ring buffer is a single buffer in the memory, the interrupts sent on behalf of the target are simplified to be sent every time the hardware writes data and/or header to the memory (depending on if Target interrupts are enabled and if global MSIen is set).

The conditions which result in a status header being written are further governed by these policies:

- TCTRL.SCHWBP: upon successful completion of target cycles.
- TCTRL.UCHWBP: upon unsuccessful completion of target cycles.
- TCTRL.URxTWP: upon unsuccessful completion of writes to target.

This simplified model is used since the hardware supports multiple logical devices and transactions of each type are pushed to memory by the hardware in a sequential manner. The firmware then has to parse each header to determine what each transaction is before the firmware takes appropriate action.

A single Target Interrupt Enable bit is maintained in MMIO space (TCTRL.TIE) which gates the sending of this interrupt. An associated status bit (TSTS.TIS) is also maintained in the hardware.

Similar to master interrupts, target interrupts are also serialized and ordered. The decision to send MSI is made after the hardware has written back the header for the received transaction to the memory ring buffer.

1. Receive target cycle from SMBus.
2. Perform data and header WB to memory.
3. Write 1 to the appropriate Cause Status bit (TSTS.TIS).

If TCTRL.TIE is enabled (locally and globally), MSI is sent and the cause status is auto cleared else MSI is not sent and cause remains set.

Note: Firmware implementation:

1. If the cause is set for a previous target transaction written to memory and the firmware enables the global (and local) interrupt enable, the hardware does not send an interrupt for a previously set cause.
2. It is the responsibility of the firmware to ensure that if the cause is set, the previous transactions are accounted for before enabling MSI (globally and locally).

15.5.3 Error Interrupts

The hardware tracks the following conditions for errors:

1. SMBus clock-low time-out: status is set when the hardware observes the SMBus clock asserted low for more than the value programmed by the firmware.
2. SMBus data-low time-out: status is set when the hardware observes the SMBus data asserted low for more than the value programmed by the firmware.
3. Target ring buffer almost full: status is set when the hardware detects the target ring buffer has less than 85-B free space remaining in the target ring buffer. This check is done after the hardware performs a memory write for data/header.
4. Target ring buffer full: status is set when the hardware in unable to evict its internal buffer/header WB to memory due to lack of space in the target ring buffer. This check is done every time the hardware needs to write to memory.

Table 15-22. Error MSI Scheduling

Event	Global MSIen	Cause Interrupt Enable[1]	Cause Interrupt Status[2]	MSI Action
Cause occurs, but interrupts are not enabled	0	0	0 -> 1	No MSI sent
MSI enable is set, cause enable and cause status are previously set	0 -> 1	1	1	Send MSI
MSI enable and cause status previously set, cause interrupt enable is set	1	0 -> 1	1	Send MSI
Cause status register gets set, interrupts are enabled	1	1	0 -> 1	Send MSI
Cause status register previously set, interrupts are enabled, new cause occurs	1	1	1 -> 1	Does not occur since if interrupts are enabled, Cause Status is cleared. If the clearing happens on the same clock as the new cause is set, it is the rule in the next row before.
New cause occurs in the same clock as the previous MSI scheduled was sent	1	1	1 -> 1	Send MSI
Scheduled MSI was sent	1	1	1 -> 0	The hardware auto-clears the cause register.
MSI is scheduled to be sent when MSI enable is cleared	1 -> 0	1	0 -> 1 or 1 -> 1	The hardware schedules the MSI or drops it. Any pending MSIs must be reflected in the MSI Pending Register.

1. Cause Interrupt Mask: see ERRINTMSKand ERRAERMSK.
2. Cause Interrupt Status: see ERRSTS.

15.5.4 Interrupt Cause Logging

Before sending any MSI, the hardware writes information pertinent to the interrupt cause to a memory location identified by Table 15-23.

As indicated in Table 15-23, the Dwords of SMTICL are divided into three categories. The hardware updates only the relevant Dword depending on the nature of the interrupt: interrupts generated during a master transaction involves updates only to Dword #0; likewise target transactions involve only Dword #1. However, if an error condition is present, the hardware requires also writing to Dword #2 to capture the nature of the error.

The firmware always reads all three Dwords, and upon reading them clears any Dword which has non-zero data. (At a minimum two Dwords must be read—master + error or target + error.)

Table 15-23. Interrupt Cause Information

Dword	Bit	Field	Description
Information from MASTER			
0	31	MSTR.VALID	Master interrupt cause is valid 1: denotes master status (MSTS), including the master hardware tail pointer, has been written
0	30:0	Reserved	
Information from TARGET			
1	31	TRGT.VALID	Target interrupt cause is valid 1: denotes the target hardware head pointer (HTHP) has been written
1	30:0	Reserved	
Error Information			
2	31	ERR.VALID	Error interrupt cause is valid 1: denotes errors status (ERRSTS) has been written
2	30:0	Reserved	

15.6 SMT RAS Architecture

15.6.1 Soft Reset (DEVCTL.IFLR and GCTRL.SRST)

The SMT supports several types of soft reset including:

- Function-level reset as defined by the *PCI Local Bus Specification*, Revision 3.0 (see DEVCTL.IFLR).
- Soft reset directed to the SMT controller (see GCTRL.SRST).

In each case, the soft reset applies only to the specific SMT function addressed. Asserting a soft reset has these effects:

- An immediate and abrupt reset of the SMT Master and Target logic (which causes a protocol violation of any pending master or target cycles),
- Clear all the MMIO registers except those register bits denoted as sticky or RW-O or primary-reset-only (PRST) as listed in Table 15-24, and
- Release the SMBus Clock and Data lines.

Table 15-24. SMT Soft Reset Exceptions

Register Fields	Comments
Sticky (S)	
AERCAPCTL.FEP	
AERHDRLOG.TLPHDRLOG	
ERRCORMSK.x, x = CIEM, ANFEM	
ERRCORSTS.x, x = CIE, ANFE	
ERRUNCMSK.x, x = UIEM, UREM, MTLPEM, UCEM, CAEM, CTEM, PTLPEM	
ERRUNCSEV.x, x = UIES, URES, MTLPES, UCES, CAES, CTES, PTLPES	
ERRUNCSTS.x, x = UIE, URE, MTLPE, UCE, CAE, CTE, PTLPE	
PMCSR.PMEEN	
RID.RID	
Primary-Reset-Only (PRST), including Read-Write-Once (RW-O)	
AERCAPHDR.NCO	
CAPPTR.CPTR	
CCR.x, x = BASE, SUB, RLPI	
DEVCAP.FLR	
DEVCAP2.CTRS	
INTP.INTP	
PLKCTL.CL	
PMCSR.NSR	
SID.SID	
SVID.SVID	

Exceptions to function-level resets are listed in Table 15-25.

Table 15-25. SMT Function Level Reset Exceptions

Register Fields	Comments
DEVCTL.x, x = IFLR and MPS	In addition to any RWS, ROS, and RW1CS fields.
IOSFDEVCLKGCTL.x, x = ICGE and ICT	
SBDEVCLKGCTL.x, x = ICGE and ICT	

15.6.2 Target Reset (GCTRL.TRST)

The scope of this reset is limited to the target FSM. Upon assertion, the hardware gracefully terminates any pending transaction at the next byte boundary and build a descriptor which the firmware rejects.

15.7 MCTP Over SMBus Packet Header Format

This section is informational. The hardware does not inspect or decode any portion of an Management Component Transport Protocol (MCTP) packet.

MCTP messages with SMBus packet headers can encapsulate data from various protocols like ASF, Network, etc. The command code in the packet (byte 1) identifies this packet as MCTP and the firmware takes appropriate action. Refer to the *MCTP Specification* for further details on the protocol.

The following information is contained in the *MCTP Specification* but is reproduced here for completeness.

Figure 15-15. MCTP Over SMBus Packet Format

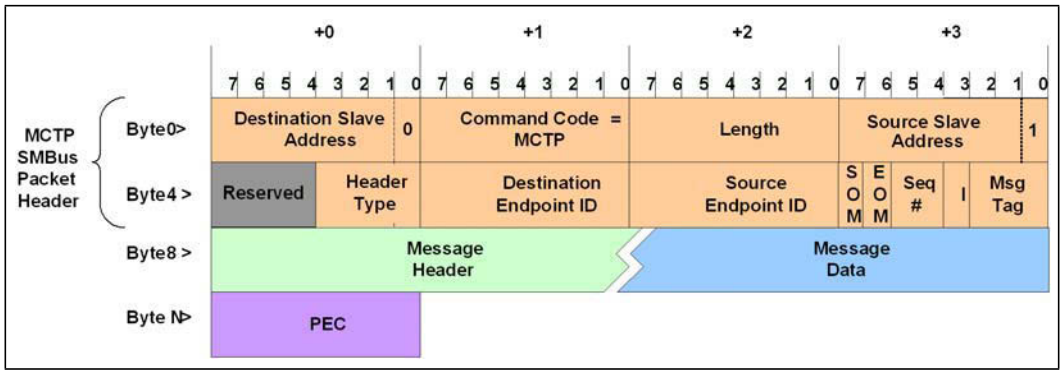


Table 15-26. MCTP Over SMBus Packet Format (Sheet 1 of 2)

Byte #	Field	Description
0	Destination Slave Address	For the local SMBus, slave address for the target device. Bit 0 is always 0 since all MCTP messages are writes.
1	Command Code	Command code indicating MCTP packet. This value is fixed at 0Fh.
2	Length	Length of the Block Write. Analogous to the Byte Count field of the SMBus Block Write except length is not limited to 32 bytes. This count indicates the number of bytes to follow in the current message without counting the PEC byte.
3	Slave Address	Bits [7:1]: For the local SMBus, the slave address of the message initiator. This initiator is a repeater and cannot be assumed to be the original initiator. Bit 0: Reserved, always 1b.
4	Header Type	Bits [7:4]: Reserved Bits [3:0]: MCTP Version: 1h for all MCTP device compliant to the *MCTP 1.0 Specification*.
5	Destination Endpoint ID	Unique ID on the System for the destination of the message. This ID reflects the final destination of the message. Reserved Values: 0 = Local Bus only. Terminate at Receiver. 1 = Route to Root Management Controller. 2-7 = Reserved for future definition.
6	Source Endpoint ID	Unique ID on the System for the originator of the message. This ID reflects the true initiator of the message.

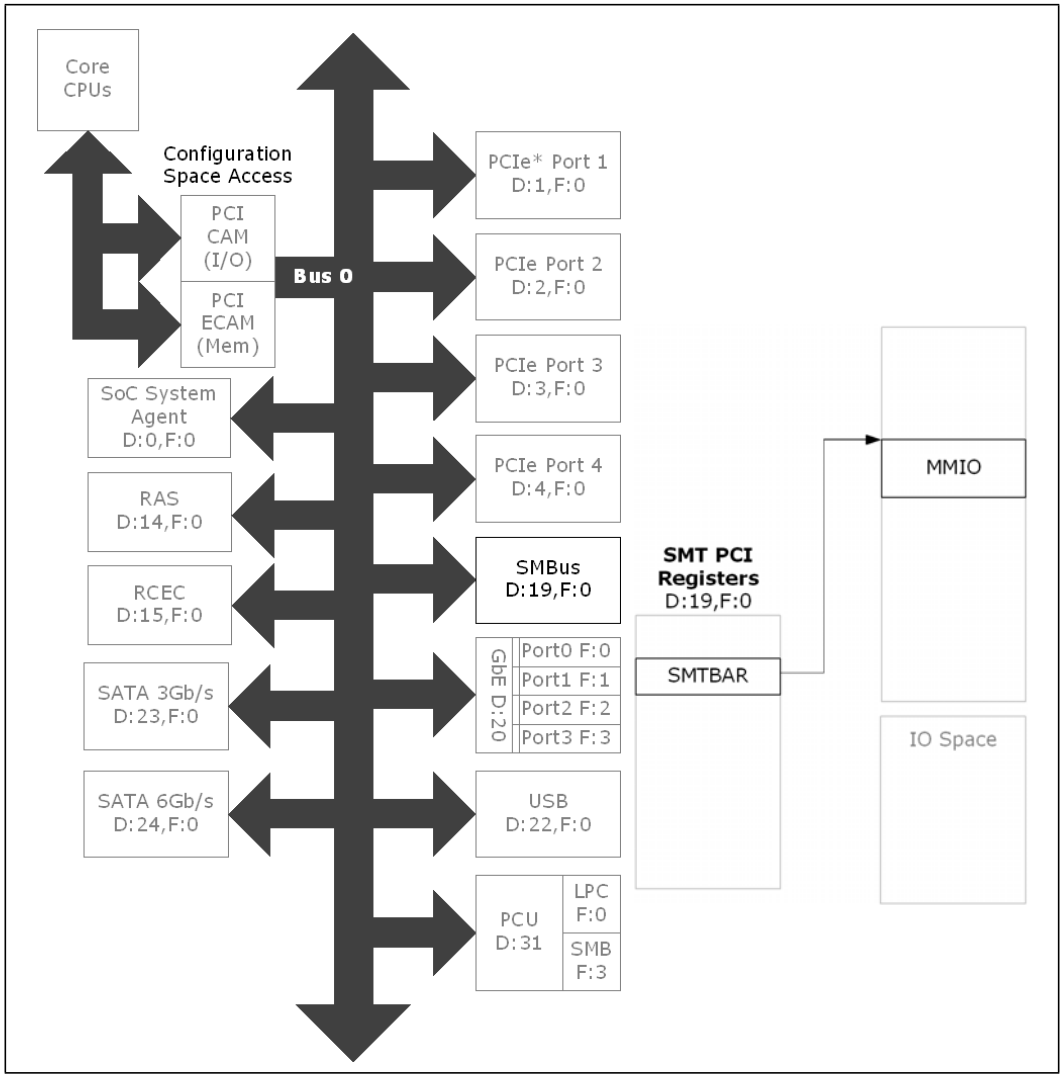
Table 15-26. MCTP Over SMBus Packet Format (Sheet 2 of 2)

Byte #	Field	Description
7	Message Control	Bit 7: **Start of Message** (SOM): Set to 1 if the packet is the start of message. Bit 6: **End of Message** (EOM): Set to 1 if the packet is the end of message. Combinations of SOM, EOM are used to identify the Start, Middle, and End of a Message that is split across one or more packets. Bits [5:4]: **Sequence Number**: For messages that span multiple packets, this field helps identify lost intermediate packets. Increments on each successive packet. The value of the Sequence Number for Packet N, is the Packet N-1 Sequence Number + 1 mod 4. Bit 3: **Initiator** (I): Set to 1 if value of Message Tag field was assigned by the source (for requests) or set to 0 of the Message Tag was assigned by the destination (for responses). Bits [2:0]: **Message Tag** (MT): This field, along with the Source Endpoint ID, identifies a unique Message ID. The Message ID must be unique for all outstanding request transactions that require a response and have not yet receive the response. For Response messages, the Message Tag sent with the Request is returned in the Response.
8	Message Header	Begin MCTP Message Header
M	Message Data	MCTP defined
N	PEC	Single byte Packet Error Check accompanies all MCTP messages.

15.8 Register Maps

Figure 15-16 shows the C2xx0 SMBus Message Transport (SMT) registers from a system viewpoint.

Figure 15-16.SMT Controller Register Map



15.8.1 Registers in Configuration Space

The SMT Controller registers in Configuration Space are shown in Table 15-27 through Table 15-32 on page 352. The registers are in the configuration space at bus 0, device 19 (decimal), function 0. The offset addresses are listed.

Table 15-27. PCI Standard Type 0 Header - SMBus Message Transport Controller

Offset	Name	Description
0x00	VID/DID	DID: 1F10h, 1F11h, 1F12h, 1F13h
0x04	PCICMD	PCI Command Register
0x06	PCISTS	PCI Status Register
0x08	RID	Revision ID Register
0x09	CCR	Class Code Register
0x0C	CLS	Cacheline Size Register
0x0D	PLAT	Primary Latency Timer
0x0E	HDR	Header Type Register
0x0F	BIST	Built-In Self-Test
0x10	SMTBAR	Memory Base Address Register
0x2C	SVID	Subsystem Vendor ID Register
0x2E	SID	Subsystem ID Register
0x34	CAPPTR	Capabilities Pointer Register
0x3C	INTL	Interrupt Line Register
0x3D	INTP	Interrupt Pin Register
0x3Eh	BCTL	Bridge Control Register

Table 15-28. PCI Express Capability - SMT Controller

Offset	Name	Description
0x040	EXPCAPLST	PCI Express* Capability List Register
0x042	EXPCAP	PCI Express Capabilities Register
0x044	DEVCAP	Device Capabilities Register
0x048	DEVCTL	Device Control Register
0x04A	DEVSTS	Device Status Register
0x064	DEVCAP2	Device Capabilities 2 Register
0x068	DEVCTL2	Device Control 2 Register
0x06A	DEVSTS2	Device Status 2 Register

Table 15-29. Message Signaled Interrupts (MSI) Capability - SMT Controller

Offset	Name	Description
0x080	PMCAPLST	Power Management Capability List Register
0x082	PMCAP	Power Management Capabilities Register
0x084	PMCSR	Power Management Control/Status Register

Table 15-30. Message Signaled Interrupts (MSI) Capability

Offset	Name	Description
0x08C	MSICAPLST	MSI Capability List Register
0x08E	MSICTL	MSI Message Control Register
0x090	MSIADDR	MSI Message Address Register
0x098	MSIDATA	MSI Message Data Register
0x09C	MSIMSK	MSI Mask Bit Register
0x0A0	MSIPENDING	MSI Pending Bit Register

Table 15-31. Advanced Error Reporting (AER) Extended Capability - SMT Controller

Offset	Name	Description
0x100	AERCAPHDR	Advanced Error Reporting Extended Capability Header
0x104	ERRUNCSTS	Uncorrectable Error Status Register
0x108	ERRUNCMSK	Uncorrectable Error Mask Register
0x10C	ERRUNCSEV	Uncorrectable Error Severity Register
0x110	ERRCORSTS	Correctable Error Status Register
0x114	ERRCORMSK	Correctable Error Mask Register
0x118	AERCAPCTL	Advanced Error Capabilities and Control Register
0x11C - 0x128	AERHDRLOG[1-4]	Header Log Register

Table 15-32. Device-Specific Registers

Offset	Name	Description
0xEA	PLKCTL	Personality Lock Key Control Register

15.8.2 Registers in Memory Space

The SMT Controller registers in Memory Space are shown in Table 15-33. These Memory-Mapped I/O registers begin at the address specified by the 64-bit register SMbus Base Address Register (SMTBAR) at offset 10h in the configuration space at bus 0, devices 19 (decimal), function 0. The offset addresses are listed and must be accessed by 32-bit memory transactions on the PC Express* interface or by local CPU.

Table 15-33. Memory Space Address and Description (Sheet 1 of 2)

Address Offset	Register Description and Name
General Registers	
000h	General Control Register (GCTRL)
004h	Reserved
008h	SMT Interrupt Cause Location Register (SMTICL)
010h	Error Interrupt Mask Register (ERRINTMSK)
014h	Error AER Mask Register (ERRAERMSK)
018h	Error Status Register (ERRSTS)
01Ch	Error Information Register (ERRINFO)
020h - 0FCh	Reserved
Master Registers	
100h	Master Descriptor Base Address Register (MDBA)
108h	Master Control Register (MCTRL)
10Ch	Master Status Register (MSTS)
110h	Master Descriptor Size Register (MDS)
114h	Retry Policy Register (RPOLICY)
118h - 1FCh	Reserved
Target Registers - General	
200h	Target Buffer Base Address Register (TBBA)
208h	Target Control Register (TCTRL)
20Ch	Target Status Register (TSTS)
210h	Target Buffer Size Register (TBS)
214h	Reserved
218h	Hardware Target Head Pointer Register (HTHP)
21Ch	Firmware Target Tail Pointer Register (FTTP)
220h	Target Receive Control Register (TRxCTRL)
224h	Target Receive Status Register (TRxSTS)
228h	Target Address Control Register (TACTRL)
22Ch	Target Policy Register (TPOLICY)
230h - 23Ch	Reserved
Target Registers - Block Read	
240h	General Purpose Block Read Control Register (GPBRCTRL)
244h	Generic Programmable Read Data Buffer Register (GPBRDBUF)
248h - 27Ch	Reserved

Table 15-33. Memory Space Address and Description (Sheet 2 of 2)

Address Offset	Register Description and Name
Target Registers - ARP	
280h	SMT Address Resolution Protocol Control Register (SMTARPCTRL)
284h - 28Ch	Reserved
290h	UDID0 Data Register (UDID0)
298h	UDID0 Upper Data Register (UUDID0)
2A0h	UDID1 Data Register (UDID1)
2A8h	UDID1 Upper Data Register (UUDID1)
2ACh - 2BCh	Reserved
Target Registers - Reserved	
2C0h - 2FCh	Reserved
PHY Registers	
300h	SMBus PHY Global Timing Register (SPGT)
304h	SMBus PHY Master Timing Register (SPMT)
308h	SMBus PHY Slave Timing Register (SPST)
30Ch	SMBus Fair Timing Register (SMBFT)
310h	Clock Low Time-out Control Register (CLTC)
314h - 37Ch	Reserved
Debug Registers	
380h	Dynamic Clock Gating Register (DCLKGT)
384h	SUS Well Chicken Bits Register (SUSCHKB)
388h	Debug Control Register (DBCTRL)
38Ch	Debug Status Register (DBSTS)
390h and remaining	Reserved

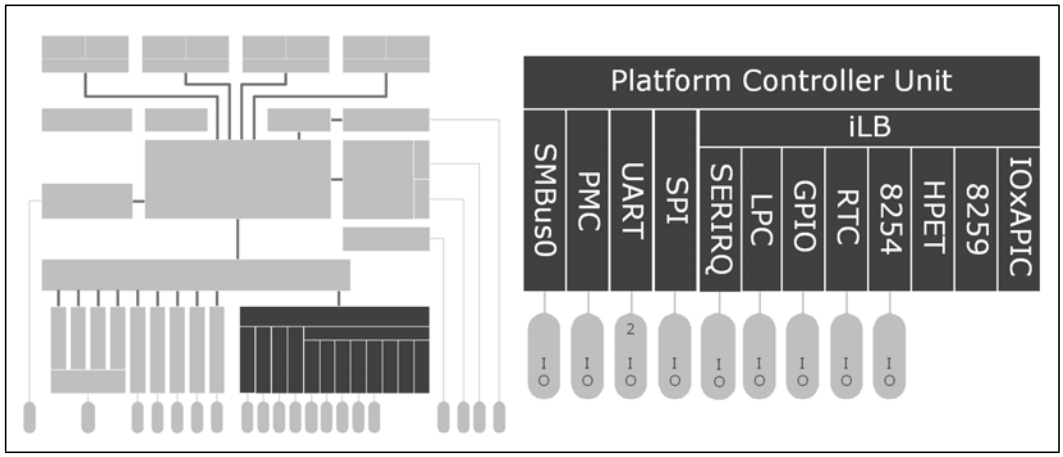
§ §

16 Platform Controller Unit (PCU)

The Platform Controller Unit (PCU) is a collection of hardware blocks that are critical for operation including x86 legacy compatibility that is required for Windows*. These hardware blocks include the Intel Legacy Block (iLB), SMBus, UART, SPI and most of the Power Management Controller (PMC).

The PCU also implements some high-level configuration features for the BIOS/EFI boot.

Figure 16-1. Platform Controller Unit Covered in This Chapter



16.1 Features

The features of the key PCU blocks are:

- Intel Legacy Block (iLB)
  - Discovered by the software at bus 0, device 31 (decimal), function 0 in the configuration address space
  - Supports legacy PC platform features
  - Sub-blocks include LPC, General Purpose I/O (GPIO), 8259 PIC, IOxAPIC, 8254 timers, HPET timers and the RTC
- SMBus Host Controller
  - Discovered by the software at bus 0, device 31 (decimal), function 3 in the configuration address space
  - Supports *System Management Bus (SMBus) 2.0 Specification*
  - No support for SMBus slave functionality aside from the Host Notify command
  - No Total Cost of Ownership (TCO) feature support
- Universal Asynchronous Receiver/Transmitter (UART)
  - Two UART interfaces available: UART0 (COM1) and UART1 (COM2)
  - 16550 controller compliant
  - Reduced signal count: TX and RX only
  - COM1 interface using eight I/O addressed mapped registers (0x3F8-0x3FF)
  - COM2 interface using eight I/O addressed mapped registers (0x2F8-0x2FF)
- Serial Peripheral Interface (SPI)
  - Uses memory-mapped I/O. The base address is in the iLB Configuration registers.
  - For one or two SPI Flash, of up to 16-MB size each, only. No other SPI peripherals are supported.
  - Stores the boot firmware and system configuration data
  - Supports frequencies of 20 MHz (default) and 33 MHz
- Power Management Controller (PMC)
  - Uses the memory-mapped I/O. The base address is in the iLB Configuration registers.
  - Uses the I/O-space-mapped I/O. The base address is in the iLB Configuration registers.
  - Controls many of the power management and reset features present in the SoC

16.2 Pin-Based (Hard) Straps

Strapping is a hardware mechanism used for system configuration control. Some of the functional signal pins are also defined as SoC strapping pins. These pins are sampled at various reset points to select configuration information. Each strapping pin is briefly re-configured as an SoC input, and its state is sampled and latched by the SoC hardware. After sampling, the SoC pin is configured in its normal functional characteristics.

The strapping pins are listed in Table 16-1.

Table 16-1. Hard Pin Straps (Sheet 1 of 2)

Signal Ball/Pin Name	Ball Location on SoC Package	Strap Usage	Sampled by This Reset	Internal PU/PD When Sampled	Notes
ERROR2_B	AL63	Reserved for Intel. Must be logic low during sampling.	COREPWROK	20 kΩ Pull-down	1
ERROR1_B	AL62	Reserved for Intel. Must be logic low during sampling.	COREPWROK	20 kΩ Pull-down	1
ERROR0_B	AL65	Reserved for Intel. Must be logic low during sampling.	COREPWROK	20 kΩ Pull-down	1
UART1_TXD	AH50	SPI Flash Descriptor security is overridden when the strap pin is sampled low. This pin is temporarily pulled-up internally during the sample period.	COREPWROK	20 kΩ Pull-up	1
FLEX_CLK_SE0	AH59	Boot Flash device location. LPC bus interface (if sensed low) versus SPI (sensed high). This pin is temporarily pulled-up internally during the sample period.	COREPWROK	20 kΩ Pull-up	1
FLEX_CLK_SE1	AG56	This pin is temporarily pulled-up internally during the sample period. Must be logic high during sampling.	COREPWROK	20 kΩ Pull-up	1
AR51_RSVD	AR51	Reserved for Intel.	COREPWROK	20 kΩ Pull-up	1
GPIO_SUS0	V66	Reserved for Intel. Must be logic low during sampling.	SUS Power OK	20 kΩ Pull-down	1
GPIO_SUS1	W54	If sensed low, the 2.5-GbE capability, if available, is disabled. This pin must be sampled high for the 2.5-GbE capability to function. This pin is temporarily pulled-down internally during the sample period.	SUS Power OK	20 kΩ Pull-down	2
GPIO_SUS2	T53	Reserved for Intel.	SUS Power OK	None	1
CPU_RESET_B	Y63	Reserved for Intel.	SUS Power OK	None	1
PMU_SUSCLK	AD58	Reserved for Intel. Must be logic low during sampling.	SUS Power OK	20 kΩ Pull-down	1
PMU_PLTRST_B	AE62	Reserved for Intel. Must be logic low during sampling.	SUS Power OK	20 kΩ Pull-down	1
SUS_STAT_B	AB65	Reserved for Intel. Must be logic low during sampling.	SUS Power OK	20 kΩ Pull-down	1
SPI_CS0_B	Y65	After G3 Enable (AG3E) strap. This strap is used for the first G3 boot. If sensed low, the system transitions to the S0 state (boot) when power is applied. If sensed high, the system transitions to the S5 state (Soft Off) when power is applied. In the S5 state, the only enabled wake-up event is the power button (PMU_PWRBTN_B) or any enabled wake event that was preserved through a power failure.	SUS Power OK	20 kΩ Pull-up	1

Table 16-1. Hard Pin Straps (Sheet 2 of 2)

Signal Ball/Pin Name	Ball Location on SoC Package	Strap Usage	Sampled by This Reset	Internal PU/PD When Sampled	Notes
SPI_CS1_B	AC58	Reserved for Intel. Must be logic low during sampling.	SUS Power OK	20 kΩ Pull-down	1
NCSI_RXD1	V63	When the NCSI_RXD1 pin is sensed as a logic high state at the SUS power-ok time, the Ethernet controller is enabled (internally powered) during the S0 state and the S5 state. Otherwise, the Ethernet controller is enabled only in the S0 state. ***Note:*** This pin is temporarily pulled-up internally during the sample period. When the integrated Ethernet controller is powered-on during S5, the Wake-on-LAN and other LAN management capabilities can be utilized.	SUS Power OK	20 kΩ Pull-up	3
NCSI_ARB_OUT	Y59	When sampled low, this indicates the GBE_SMBus is used in the platform board design for management. When sampled high, this indicates the NC-SI is to be used for management. ***Note:*** This pin is temporarily pulled-down internally during the sample period.	SUS Power OK	20 kΩ Pull-down	4

Notes:
1. Once the pin-strap sampling period is over, the pin is configured with its normal functional characteristics.
2. Once the pin-strap sampling period is over, if the NCSI_ARB_OUT strap pin is sampled as a low, the pin is configured as an input, GPIO_SUS1. If the NCSI_ARB_OUT strap pin is sampled as a high, the pin is configured as an input, NCSI_RXD0.
3. Once the pin-strap sampling period is over, this pin becomes an output, NCSI_RXD1, regardless how NCSI_ARB_OUT was sampled.
4. Once the pin-strap sampling period is over, if the NCSI_ARB_OUT strap pin is sampled as a low, the pin is configured as an output, Y59_RSVD. If the NCSI_ARB_OUT strap pin is sampled as a high, the pin is configured as an input, NCSI_ARB_OUT.

For the hard strap pins, the customer board may need to provide external pull-down and pull-up resistors to force the pin to the intended state of the strapping mechanism.

The pins sampled by COREPWROK are done so when COREPWROK is asserted. Before and during the assertion, these pins are temporarily SoC input pins and, if indicated, have an internal temporary termination resistor applied only during the sampling period. The hard pin-strap information must be held valid at the SoC inputs for a minimum of 400 ns after COREPWROK is asserted.

The pins sampled by SUS Power OK are done so when the active-low RSMRST_B signal goes high (deasserted). Before and during the reset deassertions, these pins are temporarily SoC input pins and, if indicated, have an internal temporary termination resistor applied only during the sampling period. The hard pin-strap information must be held valid at the SoC inputs for a minimum of 400 ns after RSMRST_B (active low) is de-asserted.

- Core-Well Strap-pin hold time required after COREPOWER driven high = 400 ns minimum
- SUS-Well Strap-pin hold time required after RSMRST_B (active low) driven high = 400 ns minimum

All of these hard strap pins return to their native, functional signal characteristics after the sampling period.

The strapping pins marked as Reserved for Intel do not need any special attention from the board designer for proper operation as long as the internal pull-up/pull-down is not defeated by the platform board design during the sampling period.

These board settings of the strapping mechanism are overwritten by the DFX Tap of the SoC. The Intel debug and test software reads and retains the board settings before the DFX Tap overwrite.

16.3 Multi-Functional Signal Pins

Besides pins that are also hard strap pins, there are other SoC pins that require special treatment by the platform board design.

16.3.1 Pins with More Than One Native Function

Three of the multi-function signal pins have more than one normal, native usage. Each also has the option of being configured as a Customer General Purpose I/O (GPIO). Usually this is done by the BIOS.

Table 16-2 lists these multi-function pins. Each native signal pin listed in the table has an internal 20-kΩ pull-up resistor to 3.3V.

Customer GPIO signal-pin details are in Chapter 25, "General-Purpose I/O (GPIO)." None of the three signals are hard pin-strap pins.

Table 16-2. Signal Pins May Require a Change to the Pin Function Code

SoC Ball/Pin Number	Power Well	Pin Function = 0 (Default)		Can be Set to Pin Function = 2		If Neither Function Used, Signal Pin Can be Used as Customer GPIO
AN65	3.3V Core	SMB_DATA2	I/O OD	UART0_RXD	I/O	GPIOS_13
AR65	3.3V Core	SMB_CLK2	I/O OD	UART0_TXD	O	GPIOS_14
AR63	3.3V Core	SMB_CLK1	I/O OD	SPKR	O	GPIOS_12

To use a pin function that is labeled as Pin Function = 2, its 32-bit Pad Configuration 0 (PCONF0) register in memory space must be properly set by the software, usually the BIOS. The 3-bit Functional Pin Multiplexer (FUN_PIN_MUX) field, bits [2:0] of the pins PCONF0 register must be set to 010 binary. The three 32-bit registers are shown in Table 16-3 on page 360 along with the register offset from IO_CONTROLLER_BASE_ADDRESS (IOBASE). IOBASE is located in the SoC configuration space at bus 0, device 31 (decimal), function 0, offset 0x04C.

Table 16-3. PCONF0 Registers to Assign Pin Function = 2

SoC Ball/Pin Number	Desired Signal for Ball/Pin	32-Bit PCONF0 Register in Memory Space	Offset From IOBASE	Set FUN_PIN_MUX to:
AN65	UART0_RXD	CFIO_REGS_PAD_SMB_DATA2_PCONF0	0x0160	2
AR65	UART0_TXD	CFIO_REGS_PAD_SMB_CLK2_PCONF0	0x0170	2
AR63	SPKR	CFIO_REGS_PAD_SMB_CLK1_PCONF0	0x01B0	2

To correctly set the PCONF0, the software must first read the 32-bit PCONF0 register, alter only bits [2:0] to 010 binary, and write the altered results to the original PCONF0 register. The original PCONF0 bits [31:3] must not be disturbed.

16.3.2 Pins of the Ethernet NC-SI Interface

There are a number of multi-function signal pins that provide the Network Controller Sideband Interface (NC-SI) signal to the SoC for the integrated Ethernet controller. Nine of these signals are shared with other signal-pin functions that can be used if the NC-SI is not used. See Table 16-4.

The NC-SI interface signals are applied to the balls/pins when the NCSI_ARB_OUT pin (Y59) is sampled high by the SoC when the hard pin-straps are sampled. See Section 16.2, "Pin-Based (Hard) Straps" on page 357. When the hard pin-strap NCSI_ARB_OUT pin is sampled low, other signals are assigned to these pins. Unlike the three multi-functional pins mentioned in the previous subsection, no software configuration of registers is needed to select the NC-SI interface.

Refer to section 2 of *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)* for a complete pin list and descriptions.

Table 16-4. Multi-Functional Signal Pins Controlled by a Hard Pin-Strap

SoC Ball/Pin Number	Power Well	After Hard Pin-Strap NCSI_ARB_OUT Sampled as Low		After Hard Pin-Strap NCSI_ARB_OUT Sampled as High		If Neither Function Used, Signal Pin Can be Used as Customer GPIO
T59	3.3V SUS	GBE_SMBD	I/O OD	NCSI_TX_EX	I	No
P50	3.3V SUS	GBE_SMBCLK	I/O OD	NCSI_CLK_IN	I/O OD	No
T55	3.3V SUS	GBE_SMBALRT_N	I/O OD	NCSI_CRS_DV	O	No
T48	3.3V SUS	GBE_SDP0_1[1]	I	NCSI_ARB_IN	I	GPIO_SUS18
Y53	3.3V SUS	GBE_MDIO1_I2C_DATA	I/O OD	NCSI_TXD0	I	GPIO_SUS27
Y54	3.3V SUS	GBE_MDIO1_I2C_CLK	I/O OD	NCSI_TXD1	I	GPIO_SUS26
W54	3.3V SUS	GPIO_SUS1[2]	I	NCSI_RXD0	O	GPIO_SUS1
V63	3.3V SUS	Reserved for Intel Use	N/A	NCSI_RXD1	O	GPIO_SUS23
Y59	3.3V SUS	Y59_RSVD	O	NCSI_ARB_OUT	O	No

1. When the BIOS starts, GBE_SDP0_1 is an input. It can be re-configured to be an output.
2. When the BIOS starts, GPIO_SUS1 is an input. It can be re-configured to be an output.

16.4 Soft Straps

The following section provides details related to the storage and configuration of soft straps which are used to determine the native function for each specific capability.

Soft strap information is stored within the Flash Descriptor region 0 of the SPI firmware image. The start address of the soft strap definition space is located within the Flash Map 1 register within the Flash Strap Base Address Register (FLMAP1[23:16]) field referred to as FISBA. See Section 22.5, “Flash Descriptor” on page 476.

16.4.1 Flash Descriptor Soft Strap Definition

The default value represents the internal strap signal value used if the SPI Flash is not valid.

At boot, the SPI controller reads the soft strap content from the SPI Flash and then provides this soft strap content to the various SoC controllers.

Table 16-5. Flash Descriptor Soft Strap (Sheet 1 of 10)

FITC SoC Strap Number	FISBA + Offset	Bit Offset	Soft Strap Name	Description	Default
0	+ 0h	1:0	BBS	SPI Boot Block Size: 2'b00: 64 KB (Default) 2'b01: 128 KB 2'b10: 256 KB 2'b11: Reserved	2'b00
0	+ 0h	2	Intel® QuickAssist Technology Disable (SKU Specific)	Intel® QuickAssist Technology (B0:D11) Disable: 1'b0 (false) - Enable 1'b1 (true) - Disable ***Note:*** Has no effect if the SKU does not have Intel® QuickAssist Technology	1'b0
0	+ 0h	3	SATA 3 Disable	SATA 3 (B0:D24) Disable: 1'b0 (false) - Enable 1'b1 (true) - Disable ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 6 SATA 3 Power Enable Lane 0 • SoC Strap 6 SATA 3 Power Enable Lane 1 • SoC Strap 6 SATA 3 Power Enable	1'b0
0	+ 0h	4	Reserved	Reserved	1’b0
0	+ 0h	5	GbE Port 1 Disable	GbE Port 1 (B0:D20:F1) Disabled: 1'b0 (false) - Enable 1'b1 (true) - Disable	1'b0
0	+ 0h	6	GbE Port 2 Disable	GbE Port 2 (B0:D20:F2) Disabled: 1'b0 (false) - Enable 1'b1 (true) - Disable	1'b0
0	+ 0h	7	GbE Port 3 Disable	GbE Port 3 (B0:D20:F3) Disabled: 1'b0 (false) - Enable 1'b1 (true) - Disable	1'b0

Table 16-5. Flash Descriptor Soft Strap (Sheet 2 of 10)

FITC SoC Strap Number	FISBA + Offset	Bit Offset	Soft Strap Name	Description	Default
0	+ 0h	8	GbE ALL Disable	All GbE Ports Disabled: 1'b0 (false) - Enabled 1'b1 (true) - Disabled Setting this bit overrides individual GbE port disable bits.	1'b0
0	+ 0h	9	SATA 2 Disable	SATA 2 (B0:D23) Disable: 1'b0 (false) - Enabled 1'b1 (true) - Disabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 6 SATA 2 Power Enable Lane 0 • SoC Strap 6 SATA 2 Power Enable Lane 1 • SoC Strap 6 SATA 2 Power Enable Lane 2 • SoC Strap 6 SATA 2 Power Enable Lane 3 • SoC Strap 6 SATA 2 Power Enable	1'b0
0	+ 0h	11:10	Reserved	Reserved	2'b00
0	+ 0h	12	PCIe RP1 Disable	PCIe* Root Port 1 (B0:D1) Disabled: 1'b0 (false) - Enabled 1'b1 (true) - Disabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 5 PCIe Lane Power Enable 0 • SoC Strap 5 PCIe Lane Power Enable 1 • SoC Strap 5 PCIe Lane Power Enable 2 • SoC Strap 5 PCIe Lane Power Enable 3 • SoC Strap 8 PCIe RP1 (B0:D1) Disable	1'b0
0	+ 0h	13	PCIe RP2 Disable	PCIe Root Port 2 (B0:D2) Disabled: 1'b0 (false) - Enabled 1'b1 (true) - Disabled ***Note:*** If PCIe Root Port 2 is disabled, PCIe Root Port 1 must be disabled as well. ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 5 PCIe Lane Power Enable 0 • SoC Strap 5 PCIe Lane Power Enable 1 • SoC Strap 5 PCIe Lane Power Enable 2 • SoC Strap 5 PCIe Lane Power Enable 3 • SoC Strap 5 PCIe Lane Power Enable 4 • SoC Strap 5 PCIe Lane Power Enable 5 • SoC Strap 5 PCIe Lane Power Enable 6 • SoC Strap 5 PCIe Lane Power Enable 7 • SoC Strap 8 PCIe RP2 (B0:D2) Disable	1'b0

Table 16-5. Flash Descriptor Soft Strap (Sheet 3 of 10)

FITC SoC Strap Number	FISBA + Offset	Bit Offset	Soft Strap Name	Description	Default
0	+ 0h	14	PCIe RP3 Disable	PCIe Root Port 3 (B0:D3) Disabled: 1'b0 (false) - Enabled 1'b1 (true) - Disabled ***Note:*** If PCIe Root Port 3 is disabled, PCIe Root Port 4 must be disabled as well. ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 5 PCIe Lane Power Enable 8 • SoC Strap 5 PCIe Lane Power Enable 9 • SoC Strap 5 PCIe Lane Power Enable 10 • SoC Strap 5 PCIe Lane Power Enable 11 • SoC Strap 5 PCIe Lane Power Enable 12 • SoC Strap 5 PCIe Lane Power Enable 13 • SoC Strap 5 PCIe Lane Power Enable 14 • SoC Strap 5 PCIe Lane Power Enable 15 • SoC Strap 8 PCIe RP3 (B0:D3) Disable	1'b0
0	+ 0h	15	PCIe RP4 Disable	PCIe Root Port 4 (B0:D4) Disabled: 1'b0 (false) - Enabled 1'b1 (true) - Disabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 5 PCIe Lane Power Enable 12 • SoC Strap 5 PCIe Lane Power Enable 13 • SoC Strap 5 PCIe Lane Power Enable 14 • SoC Strap 5 PCIe Lane Power Enable 15 • SoC Strap 8 PCIe RP4 (B0:D4) Disable	1'b0
0	+ 0H	16	Reserved	Reserved	1'b0
0	+ 0h	17	GbE Powered in S5	GbE functionality available in S5 system state: 1'b0 (false) - Not available 1'b1 (true) - Available	1'b0
0	+ 0h	19:18	Reserved	Reserved	3'b000
0	+ 0h	20	VDDQ Channel 0 Enable	Channel 0 DRAM SVID VDDQ Voltage Enabled 1'b0 (false) - Disabled (No SVID Support) 1'b1 (true) - Enabled (With SVID Support) 1. SVID based VR on VDDQ0 and SVID based VR on VDDQ1 • Soft straps should be set to VDDQ CH0 : 1 , VDDQ CH1 : 1 2. Single SVID Based VR for both VDDQ0 and VDDQ1: • Soft straps should be set to VDDQ CH0 : 1 , VDDQ CH1 : 0 3. No SVID based VR for both VDDQ0 and VDDQ1: • Soft straps should be set to VDDQ CH0 : 0 , VDDQ CH1 : 0	1'b1

Table 16-5. Flash Descriptor Soft Strap (Sheet 4 of 10)

FITC SoC Strap Number	FISBA + Offset	Bit Offset	Soft Strap Name	Description	Default
0	+ 0h	21	VDDQ Channel 1 Enable	Channel 1 DRAM SVID VDDQ Voltage Enabled 1'b0 (false) - Disabled (No SVID Support) 1'b1 (true) - Enabled (With SVID Support) 1. SVID based VR on VDDQ0 and SVID based VR on VDDQ1 • Soft straps should be set to VDDQ CH0 : 1 , VDDQ CH1 : 1 2. Single SVID Based VR for both VDDQ0 and VDDQ1: • Soft straps should be set to VDDQ CH0 : 1 , VDDQ CH1 : 0 3. No SVID based VR for both VDDQ0 and VDDQ1: • Soft straps should be set to VDDQ CH0 : 0 , VDDQ CH1 : 0	1'b1
0	+ 0h	26:22	Reserved	Reserved	5'h0
0	+ 0h	27	No Reboot	Platform Reset (PMU_PLTRST_B) after TCO WDT second time expiration Disabled: 1'b0 (false) - Enabled 1'b1 (true) - Disabled	1'b0
0	+ 0h	31:28	Reserved	Reserved	4'h0
1	+ 4h	3:0	Reserved	Reserved	N/A
1	+ 4h	4	COREPWROK_Wait	Wait Forever for Core Power OK (COREPWROK) assertion: 1’b0 - Watch Dog Timer (WDT) expiration will cause an SoC reset 1’b1 -WDT will not cause an SoCSoC reset while waiting for Core Pwr OK	1’b0
1	+ 4h	31:5	Reserved	Reserved	N/A
2	+ 8h	4:0	Reserved	Reserved	5'h0
2	+ 8h	17:5	BIOS Protected Range 4 Base	Specifies the lower base of the BIOS protected range number 4. Address bits [11:0] are assumed to be 12'h000 for the base comparison. [Goes to bits [12:0] at register: [Protected_Range_4] PR4 (at 0x84)]	13'h0
2	+ 8h	30:18	BIOS Protected Range 4 Limit	Specifies the upper limit of the BIOS protected range number 4. Address bits [11:0] are assumed to be 12'hFFF for the limit comparison. [Goes to bits [28:16] at register: [Protected_Range_4] PR4 (at 0x84)]	13'h0
2	+ 8h	31	BIOS PR4 Write Protection Enable	When set (true), this bit indicates that the base and limit fields are valid and that writes directed to the addresses between them (inclusive) must be blocked by the hardware. The base and limit fields are ignored when this bit is cleared. Disabling this protected range is done also by the security override pin strap. [This soft strap and the security override pin strap are reflected into bit 31 at register: [Protected_Range_4] PR4 (at 0x84).]	1'b0
3	+ 0Ch	1:0	Reserved	Reserved	2’b00
3	+ 0Ch	9:2	8 bit OEM Scratch Pad	8 bit OEM Scratch Pad. An example usage would to be to store system memory down information.	8’h0
3	+ 0Ch	31:10	Reserved	Reserved	22’h0
4	+ 10h	31:0	Reserved	Reserved	N/A

Table 16-5. Flash Descriptor Soft Strap (Sheet 5 of 10)

FITC SoC Strap Number	FISBA + Offset	Bit Offset	Soft Strap Name	Description	Default
5	+ 14h	5:0	Reserved	Reserved	6'h0
5	+ 14h	6	PCIe RP1, 2, 3 or 4 Enabled	PCIe Root Ports 1, 2, 3, or 4 are Enabled: 1'b0 (false) - All are Disabled. 1'b1 (true) - At least 1 is Enabled. ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 0 PCIe RP1 Disable • SoC Strap 0 PCIe RP2 Disable • SoC Strap 0 PCIe RP3 Disable • SoC Strap 0 PCIe RP4 Disable	1'b1
5	+ 14h	7	PCIe Power Enabled Lane 0	PCIe Power Enabled Lane 0: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP1.	1'b1
5	+ 14h	8	PCIe Power Enabled Lane 1	PCIe Power Enabled Lane 1: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP1.	1'b1
5	+ 14h	9	PCIe Power Enabled Lane 2	PCIe Power Enabled Lane 2: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP1.	1'b1
5	+ 14h	10	PCIe Power Enabled Lane 3	PCIe Power Enabled Lane 3: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP1.	1'b1
5	+ 14h	11	PCIe Power Enabled Lane 4	PCIe Power Enabled Lane 4: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP1 and PCIe RP2.	1'b1
5	+ 14h	12	PCIe Power Enabled Lane 5	PCIe Power Enabled Lane 5: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP1 and PCIe RP2.	1'b1
5	+ 14h	13	PCIe Power Enabled Lane 6	PCIe Power Enabled Lane 6: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP1 and PCIe RP2.	1'b1
5	+ 14h	14	PCIe Power Enabled Lane 7	PCIe Power Enabled Lane 7: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP1 and PCIe RP2.	1'b1

Table 16-5. Flash Descriptor Soft Strap (Sheet 6 of 10)

FITC SoC Strap Number	FISBA + Offset	Bit Offset	Soft Strap Name	Description	Default
5	+ 14h	15	PCIe Power Enabled Lane 8	PCIe Power Enabled Lane 8: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP3 and PCIe RP4.	1'b1
5	+ 14h	16	PCIe Power Enabled Lane 9	PCIe Power Enabled Lane 9: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP3 and PCIe RP4.	1'b1
5	+ 14h	17	PCIe Power Enabled Lane 10	PCIe Power Enabled Lane 10: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP3 and PCIe RP4.	1'b1
5	+ 14h	18	PCIe Power Enabled Lane 11	PCIe Power Enabled Lane 11: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP3 and PCIe RP4.	1'b1
5	+ 14h	19	PCIe Power Enabled Lane 12	PCIe Power Enabled Lane 12: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP4.	1'b1
5	+ 14h	20	PCIe Power Enabled Lane 13	PCIe Power Enabled Lane 13: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP4.	1'b1
5	+ 14h	21	PCIe Power Enabled Lane 14	PCIe Power Enabled Lane 14: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP4.	1'b1
5	+ 14h	22	PCIe Power Enabled Lane 15	PCIe Power Enabled Lane 15: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Disable this lane if disabling PCIe RP4.	1'b1
5	+ 14h	31:23	Reserved	Reserved	9'h0
6	+ 18h	5:0	Reserved	Reserved	6'h0

Table 16-5. Flash Descriptor Soft Strap (Sheet 7 of 10)

FITC SoC Strap Number	FISBA + Offset	Bit Offset	Soft Strap Name	Description	Default
6	+ 18h	6	SATA 2 Power Enable	SATA 2 Power Enabled: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 0 SATA 2 Disable • SoC Strap 6 SATA 2 Power Enable Lane 0 • SoC Strap 6 SATA 2 Power Enable Lane 1 • SoC Strap 6 SATA 2 Power Enable Lane 2 • SoC Strap 6 SATA 2 Power Enable Lane 3	1'b1
6	+ 18h	7	SATA 2 Power Enabled Lane 0	SATA 2 Power Enabled Lane 0: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 0 SATA 2 Disable • SoC Strap 6 SATA 2 Power Enable • SoC Strap 6 SATA 2 Power Enable Lane 1 • SoC Strap 6 SATA 2 Power Enable Lane 2 • SoC Strap 6 SATA 2 Power Enable Lane 3	1'b1
6	+ 18h	8	SATA 2 Power Enabled Lane 1	SATA 2 Power Enabled Lane 1: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 0 SATA 2 Disable • SoC Strap 6 SATA 2 Power Enable • SoC Strap 6 SATA 2 Power Enable Lane 0 • SoC Strap 6 SATA 2 Power Enable Lane 2 • SoC Strap 6 SATA2 Power Enable Lane 3	1'b1
6	+ 18h	9	SATA 2 Power Enabled Lane 2	SATA 2 Power Enabled Lane 2: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 0 SATA 2 Disable • SoC Strap 6 SATA 2 Power Enable • SoC Strap 6 SATA 2 Power Enable Lane 0 • SoC Strap 6 SATA 2 Power Enable Lane 1 • SoC Strap 6 SATA 2 Power Enable Lane 3	1'b1

Table 16-5. Flash Descriptor Soft Strap (Sheet 8 of 10)

FITC SoC Strap Number	FISBA + Offset	Bit Offset	Soft Strap Name	Description	Default
6	+ 18h	10	SATA 2 Power Enabled Lane 3	SATA 2 Power Enabled Lane 3: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 0 SATA 2 Disable • SoC Strap 6 SATA 2 Power Enable • SoC Strap 6 SATA 2 Power Enable Lane 0 • SoC Strap 6 SATA 2 Power Enable Lane 1 • SoC Strap 6 SATA 2 Power Enable Lane 2	1'b1
6	+ 18h	13:11	Reserved	Reserved	3'h0
6	+ 18h	14	SATA 3 Power Enable	SATA 3 Power Enabled: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 0 SATA 3 Disable • SoC Strap 6 SATA 3 Power Enable Lane 0 • SoC Strap 6 SATA 3 Power Enable Lane 1	1'b1
6	+ 18h	15	SATA 2 Power Enabled Lane 0	SATA 3 Power Enabled Lane 0: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 0 SATA 3 Disable • SoC Strap 6 SATA 3 Power Enable • SoC Strap 6 SATA2 Power Enable Lane 1	1'b1
6	+ 18h	16	SATA 3 Power Enabled Lane 1	SATA 3 Power Enabled Lane 1: 1'b0 (false) - Disabled 1'b1 (true) - Enabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 0 SATA 3 Disable • SoC Strap 6 SATA 3Power Enable • SoC Strap 6 SATA2 Power Enable Lane 0	1'b1
6	+ 18h	31:17	Reserved	Reserved	15'h0
7	+ 1Ch	31:0	Reserved	Reserved	N/A

Table 16-5. Flash Descriptor Soft Strap (Sheet 9 of 10)

FITC SoC Strap Number	FISBA + Offset	Bit Offset	Soft Strap Name	Description	Default
8	+ 20	11:0	PCIe Slot Width	PCIe Slot Width is used to set the slot width for each PCIe Root Port. Bits [11:09] are used for PCIe RP4. Bits [08:06] are used for PCIe RP3. Bits [05:03] are used for PCIe RP2. Bits [02:00] are used for PCIe RP1. The encoding for each set of 3 bits is: 3'b000 = Physical port width 3'b001 = x1 3'b010 = x2 3'b011 = x4 3'b100 = x8 only valid for RP1 and RP3 3'b101 = x16 only valid for RP1 ***Note:*** The Bifurcation Control register (BIFCTL0) impacts these soft straps.	12'h0
8	+ 20h	15:12	Reserved	Reserved	1'b0
8	+ 20h	16	PCIe RP1 Disable	PCIe Root Port 1 (B0:D1) Disabled: 1'b0 (false) - Enabled 1'b1 (true) - Disabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 0 PCIe RP1 (B0:D1) Disable • SoC Strap 5 PCIe Lane Power Enable 0 • SoC Strap 5 PCIe Lane Power Enable 1 • SoC Strap 5 PCIe Lane Power Enable 2 • SoC Strap 5 PCIe Lane Power Enable 3	1'b0
8	+ 20h	17	PCIe RP2 Disable	PCIe Root Port 2 (B0:D2) Disabled: 1'b0 (false) - Enabled 1'b1 (true) - Disabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 0 PCIe RP2 (B0:D2) Disable • SoC Strap 5 PCIe Lane Power Enable 0 • SoC Strap 5 PCIe Lane Power Enable 1 • SoC Strap 5 PCIe Lane Power Enable 2 • SoC Strap 5 PCIe Lane Power Enable 3 • SoC Strap 5 PCIe Lane Power Enable 4 • SoC Strap 5 PCIe Lane Power Enable 5 • SoC Strap 5 PCIe Lane Power Enable 6 • SoC Strap 5 PCIe Lane Power Enable 7	1'b0

Table 16-5. Flash Descriptor Soft Strap (Sheet 10 of 10)

FITC SoC Strap Number	FISBA + Offset	Bit Offset	Soft Strap Name	Description	Default
8	+ 20h	18	PCIe RP3 Disable	PCIe Root Port 3 (B0:D3) Disabled: 1'b0 (false) - Enabled 1'b1 (true) - Disabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 0 PCIe RP3 (B0:D3) Disable • SoC Strap 5 PCIe Lane Power Enable 8 • SoC Strap 5 PCIe Lane Power Enable 9 • SoC Strap 5 PCIe Lane Power Enable 10 • SoC Strap 5 PCIe Lane Power Enable 11 • SoC Strap 5 PCIe Lane Power Enable 12 • SoC Strap 5 PCIe Lane Power Enable 13 • SoC Strap 5 PCIe Lane Power Enable 14 • SoC Strap 5 PCIe Lane Power Enable 15	1'b0
8	+ 20h	19	PCIe RP4 Disable	PCIe Root Port 4 (B0:D4) Disabled: 1'b0 (false) - Enabled 1'b1 (true) - Disabled ***Note:*** Ensure these soft straps are set to match this selection. • SoC Strap 0 PCIe RP4 (B0:D4) Disable • SoC Strap 5 PCIe Lane Power Enable 8 • SoC Strap 5 PCIe Lane Power Enable 9 • SoC Strap 5 PCIe Lane Power Enable 10 • SoC Strap 5 PCIe Lane Power Enable 11	1'b0
8	+ 20h	31:20	Reserved	Reserved	12'h0

Note: * The Flash Image Tool Creation (FITC) SoC Strap Number is referenced in Table A-1 of the *Intel® Atom™ Processor C2000 Product Family SPI Flash Programming Tools and Users Guide* - document number 519715.

16.5 Root Complex Register Block (RCRB)

The PCU contains the Root Complex Register Block (RCRB). This block is a 32-bit register that is memory-mapped I/O, with the base address in the iLB Configuration registers. The register is named the General Control and Status (GCS) and contains the BIOS configuration and status needed for the BIOS/EFI boot.

16.5.1 Boot BIOS Straps (BBS)

The BIOS/EFI is booted from either the PCU SPI interface or the iLB LPC interface. The choice of SPI or LPC is configured by the 2-bit Boot BIOS Straps (BBS) field in the General Control and Status (GCS) register (address 0x0000 in the RCRB). The configurations of the BBS are indicated in Table 16-6. During power-up, these bits are affected by the external strap pin FLEX_CLK_SE0.

Note: The BIOS/EFI boot from the LPC interface is not available when Secure Boot is enabled.

Table 16-6. BBS Configurations

Boot BIOS Straps (GCS.BBS)	Description
00	Boot from LPC
11	Boot from SPI
01 or 10	Reserved

Note: Writes to GCS.BBS are unsuccessful if the GCS.BILD bit has been set which means that the GCS.BBS setting has been locked-down by the software.

16.6 BIOS Ranges on Flash Memory Devices

16.6.1 BIOS Decode Enable for LPC and SPI

The 32-bit BIOS Decode Enable (PCIE_REG_BIOS_DECODE_EN) register enables ranges in the BIOS for decoding purposes. It is located in configuration space at bus 0, device 31 (decimal), function 0, offset 0D8h.

This register affects the BIOS decode regardless of whether the BIOS is resident on the Low Pin Count (LPC) bus interface or the Serial Peripheral Interface (SPI).

The PCU uses the enabled memory-address ranges to decode memory requests from rest of the system. Provided that other system and security conditions are met, these memory requests are passed to the devices on the LPC bus interface or SPI devices. Whether passed to the LPC or SPI depends on which has been configured as the source of the BIOS Flash memory. See Section 16.5.1, "Boot BIOS Straps (BBS)" on page 372 for details.

See Table 16-7 for the list of enable bits and associated memory-address ranges. Multiple ranges are enabled by the software. The concept of feature space does not apply to the SPI-based Flash.

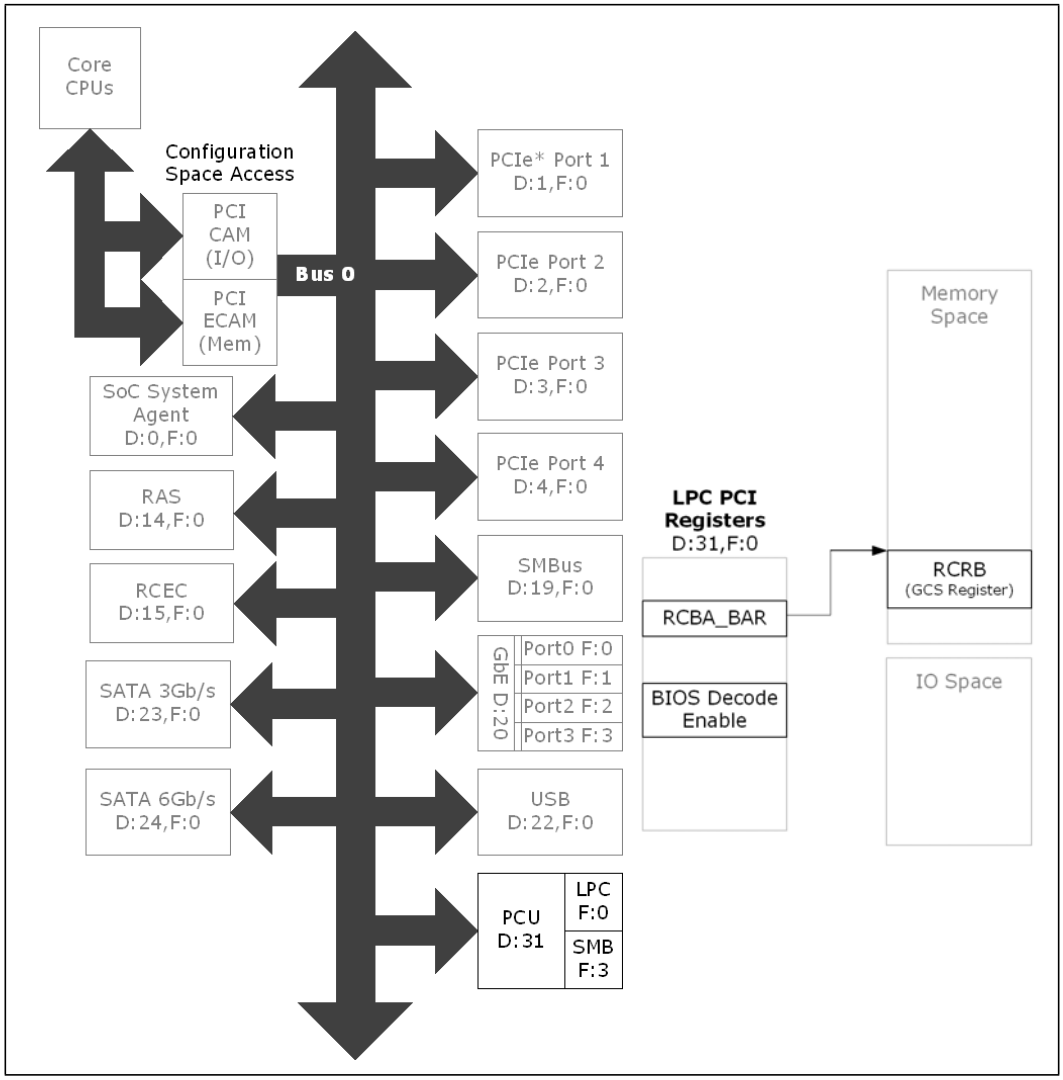
Table 16-7. Enable Bits in the BIOS Decode Enable (BDE) Register

Data Memory Address Start	Data Memory Address End	Size of Segment	Device on LPC Interface	Bit Name	Enable Bit in the BIOS Decode Enable Register	Feature Memory Address Start (LPC only)	Feature Memory Address End (LPC only)
FFF8_0000	FFFF_FFFF	512 KB	BDE.EF8	EF8	F8-FF Enable	FFB8_0000	FFBF_FFFF
FFF0_0000	FFF7_FFFF	512 KB	BDE.EF0	EF0	F0-F7 Enable	FFB0_0000	FFB7_FFFF
FFE8_0000	FFEF_FFFF	512 KB	BDE.EE8	EE8	E8-EF Enable	FFA8_0000	FFAF_FFFF
FFE0_0000	FFE7_FFFF	512 KB	BDE.EE0	EE0	E0-E7 Enable	FFA0_0000	FFA7_FFFF
FFD8_0000	FFDF_FFFF	512 KB	BDE.ED8	ED8	D8-DF Enable	FF98_0000	FF9F_FFFF
FFD0_0000	FFD7_FFFF	512 KB	BDE.ED0	ED0	D0-D7 Enable	FF90_0000	FF97_FFFF
FFC8_0000	FFCF_FFFF	512 KB	BDE.EC8	EC8	C8-CF Enable	FF88_0000	FF8F_FFFF
FFC0_0000	FFC7_FFFF	512 KB	BDE.EC0	EC0	C0-C7 Enable	FF80_0000	FF87_FFFF
FF70_0000	FF7F_FFFF	1 MB		E70	70-7F Enable	FF30_0000	FF3F_FFFF
FF60_0000	FF6F_FFFF	1 MB		E60	60-6F Enable	FF20_0000	FF2F_FFFF
FF50_0000	FF5F_FFFF	1 MB		E50	50-5F Enable	FF10_0000	FF1F_FFFF
FF40_0000	FF4F_FFFF	1 MB		E40	40-4F Enable	FF00_0000	FF0F_FFFF
Legacy Segments in the Lowest MB of Memory Space							
000F_0000	000F_FFFF	64 KB		LFE	Legacy F Segment Enable		
000E_0000	000E_FFFF	64 KB		LEE	Legacy E Segment Enable		

16.7 Register Map

Figure 16-2 shows the SoC PCU registers from a system viewpoint.

Figure 16-2. Intel® Atom™ Processor C2000 Product Family for Microserver PCU Register Map



16.7.1 PCI Configuration and Capabilities

One PCU configuration register is located in the LPC Configuration registers at bus 0, device 31 (decimal), function 0.

Table 16-8. Register Map in LPC Configuration and Capabilities

CFG Address Offset	Name	Description
0xD8	PCIE_REG_BIOS_DECODE_EN	BIOS Decode Enable
0xF0	RCRB_BASE_ADDRESS	Root Complex Register Block (RCRB) Base Address (RCBA)

16.7.2 MMIO Registers

One PCU I/O register is located in the memory space at RCRB_BASE_ADDRESS.

Table 16-9. MMIO Register Map

MEM Address	Name	Description
0x00	RCRB_GENERAL_CONTROL	General Control Status (GCS)

16.7.3 Alternate Register Access Map

The SoC maps a number of registers not exposed typically through PCI Configuration Space by an alternate mechanism that maps the registers into MMIO. The following areas are provided through this alternate access method.

Table 16-10. Alternate Access Map (Sheet 1 of 2)

Source				Destination				Description
BAR Name	Space	Start Address	End Address	Block Name	Space	Start Address	End Address	
IOBASE	MMIO	0x00000	0x01000	CORE_IO	Private CFG	0x04000	0x04FFF	CORE CFIO Controller memory space
IOBASE	MMIO	0x01000	0x017FF	SUS_IO	Private CFG	0x04000	0x04FFF	Suspend CFIO Controller memory space
GBASE	I/O	0x00000	0x0007F	CORE_IO	Private CFG	0x00000	0x0007F	CORE CFIO Controller I/O configurations
GBASE	I/O	0x00080	0x000BF	SUS_IO	Private CFG	0x00000	0x0003F	Suspend CFIO Controller I/O configurations
RCBA	MMIO	0x00200	0x003FF	USB2 bridge	Private CFG	0x00000	0x001FF	USB2 RCRB registers
MPBASE	MMIO	0x00000	0x3FFFF	SATA PHY	MMIO	0x00000	0x3FFFF	SATA PHY memory space
MPBASE	MMIO	0x40000	0x7FFFF	PCIe PHY	MMIO	0x00000	0x3FFFF	PCIe PHY memory space

Table 16-10. Alternate Access Map (Sheet 2 of 2)

Source				Destination				Description
BAR Name	Space	Start Address	End Address	Block Name	Space	Start Address	End Address	
MPBASE	MMIO	0x80000	0xBFFFF	USB2 PHY	Private CFG	0x00000	0x00000	USB2 PHY memory space
MPBASE	MMIO	0xC0000	0xFFFFF	SATA3 PHY	MMIO	0x0000	0x3FFFF	SATA3 PHY memory space
PUBASE	MMIO	0x00000	0x007FF	P-Unit	Private CFG	0x00000	0x001FF	P-Unit registers. P-Unit as opposed to HOST uses DW addressing, Hence address bits [1:0] are removed (divide by 4). BE field is still supported. All SB transactions to the P-Unit should be non-posted.

When accessing via the alternate mechanism, Intel recommends that all accesses be serialized to prevent collisions as this mechanism does not support concurrent accesses.

§ §

17 SMBus 2.0 Unit 2 - PECI

The Platform Environment Control Interface (PECI) was developed to replace I^2C as the methodology of reading CPU temperatures. The PECI specification has evolved overtime to provide a broader management interface to manage the platform. The current PECI specification is *RS - Platform Environment Control Interface (PECI) Specification,* Revision 3.0. In non-SoC environments, the PECI slave interface is implemented in the CPU complex, while the master interface is in the Platform Control Hub (PCH) and/or Base Board Management Controller (BMC). The PECI implementation only allows a single master on the bus.

On the SoC platform, the BMC acts as a PECI master and the SoC SMBus-for-PECI controller described in this chapter acts as a slave PECI proxy controller. The PECI commands are encapsulated within SMBus packets sent to the SoC. Additional, new PECI commands have been defined for the SoC, specific to its architecture.

Figure 17-1. SMBus PECI Covered in This Chapter

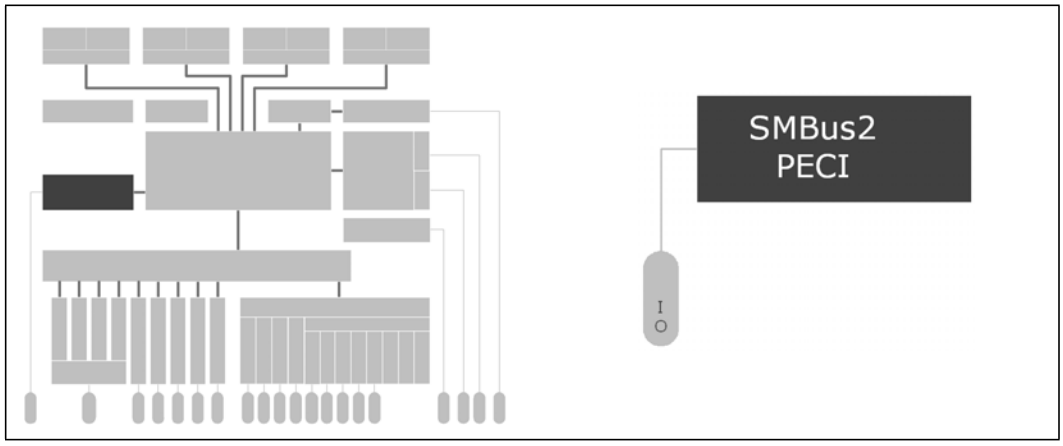


Table 17-1. References

Reference	Revision	Date	Document Title
SMBus	2.0	August 3, 2000	*System Management Bus (SMBus) Specification,* Version 2.0
PECI 3.0 Specification 31631	3.0	Mar. 15, 2012	*RS - Platform Environment Control Interface (PECI) Specification,* Revision 3.0

17.1 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

Table 17-2. Signal Names

Signal Name	Direction Type	Description
SMB_CLK2	I/OD	SMBus Clock (SMBCLK) *This signal is muxed with GPIOS_14/UART0_TXD and is used by other functions.*
SMB_DATA2	I/OD	SMBus Data (SMBDAT) *This signal is muxed with GPIOS_13/UART0_RXD and is used by other functions.*

The optional SMBus 2.0 signals, SMBALERT# and SMBSUS#, are not supported on this controller interface.

17.2 PECI over SMBus Features

The SMBus supports the external BMC, operates in slave mode only and comprehends the PECI command structure.

The PECI processes one command at time and support for multiple outstanding commands is not supported.

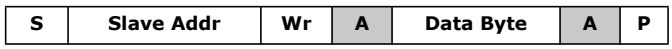
The PECI commands supported by the SoC are in Table 17-24, “Summary of DRAM Thermal Services” on page 410 and Table 17-26, “Summary of CPU Thermal and Power Optimization Services” on page 412.

17.3 SMBus Supported Transactions

In the SoC, the communication between the SMBus controller and the internal PECI bridge utilizes the SMBus block write and block read transactions as defined in the SMBus 2.0 specifications.

See Figure 17-2 to interpret the SMBus protocol drawing.

Figure 17-2. SMBus Protocol



S	Start Condition
Sr	Repeated Start Condition
Rd	Read (bit value of 1)
Wr	Write (bit value 0)
X	Shown under a field indicates that field is required To have the value of 'x'
A	Acknowledge (this bit position is 0 for an ACK or 1 for a NACK)
P	Stop Condition
PEC	Packet Error Code
…	Continuation of Protocol
	Master to Slave
	Slave to Master

17.4 SMBus Block Read/Write Transaction Formats

Figure 17-3. SMBus Block Write Command

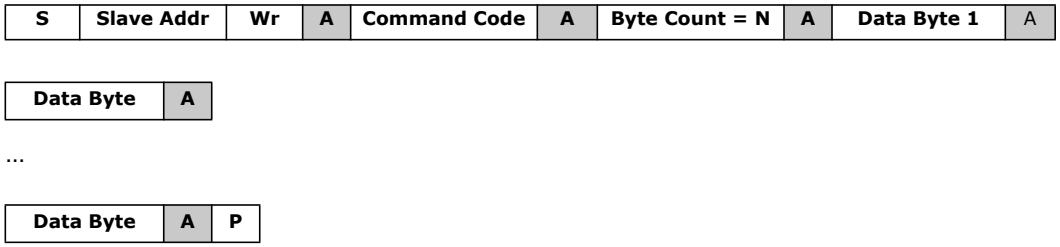
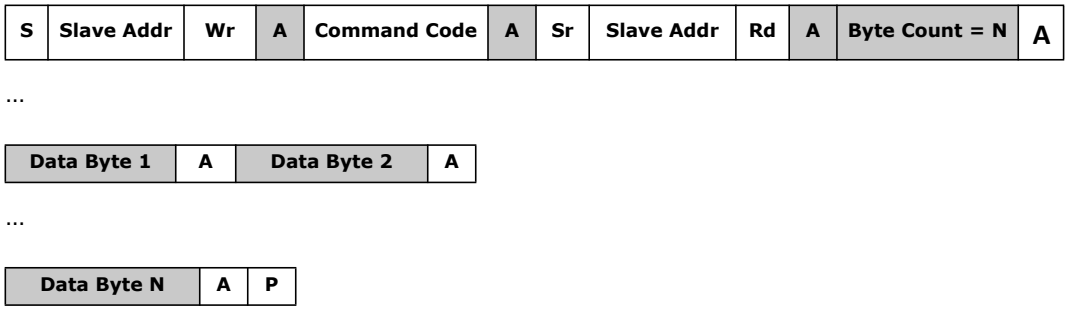


Figure 17-4. SMBus Block Read Command



17.5 SMBus Commands

Table 17-3. SMBus Write Commands

Operating Mode	Transaction	Slave Address	Command Code	Byte Count (N)	Byte 1	Byte [N:1]
PECI Mode	PECI Proxy Command	0x4c	0x62	0x4...N[1]		
PECI Mode	Reset PECI	0x4c	0x64	0		

1. See Section 17.6.2.1, "PECI Proxy Command Format" on page 385.

Table 17-4. SMBus Read Command

Operating Mode	Transaction	Slave Address (CMD Phase)	Read Command Code	Slave Address (Data Phase)	Byte Count (N)	Bytes [N:1]
PECI Mode	PECI Proxy Read Command	0x4B	0x40	0x4B[1]	0x01...N[2]	

1. Each device that exists as a slave on the SMBus has one unique seven bit address called the slave address. Each address is seven bits long with a read/write bit appended in bit position 0. When a device "sees" its address, it wakes up and responds to the rest of the command.
2. See Section 17.6.2.2, "PECI Proxy Read Command" on page 386.

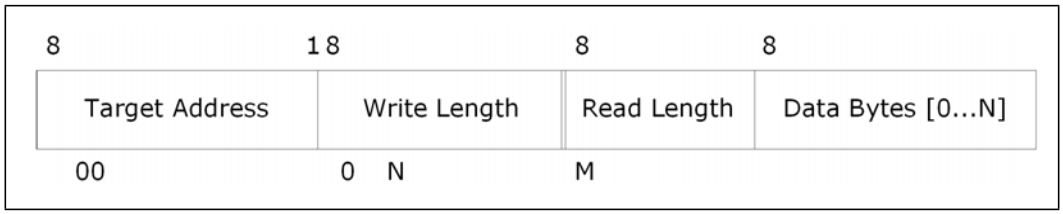
17.6 PECI Over SMBus

The PECI uses a simple structure of message header and a write-read protocol. All PECI devices have a command field. The following section describes PECI protocol. Refer to the *RS - Platform Environment Control Interface (PECI) Specification,* Revision 3.0 for details.

17.6.1 PECI Message Header in SMBus

The header conveys to the target device how many bytes master intends to send and how many it expects to receive back. The first byte of the write data is interpreted as a command to the device and must be present in all messages. The Ping() command is the only exception to this rule. Additional bytes are written to convey sub commands or to send data to the device. A zero value in the Read Length field means no data is read from the target device. See Figure 17-5 for the PECI Message Header. In the figure, N bytes are to be written to the target and M bytes are to be read back from the target.

Figure 17-5. PECI Message Header in the SMBus Packet



In the PECI Message Header, the Address Timing Negotiation (NT) and Message Timing Negotiation (MT) bits described in *RS - Platform Environment Control Interface (PECI) Specification,* Revision 3.0 are not implemented. It is the responsibility of the BMC to not include these timing bits in the header.

17.6.1.1 Target Address Field

The PECI Device Address is defined as 0x30 + a socket identifier of 0…7. In the SoC, only a single socket model is supported, thus the PECI target address is always 0x30. Other PECI Device Addresses are rejected as an errors.

17.6.1.2 Write Length Field

The Write Length byte conveys the number of bytes the originator sends the target device. Since all PECI message headers are identical in size, with the exception of Ping(), the Write Length byte only describes the number of command bytes sent. The Target Address, Write Length and Read Length fields are not included in the Write Length count. The following rules apply for the Write Length field:

- The maximum value is 0xFF.
- The command byte is included in the length.
- A Write Length of 0x0 is only for the Ping() command.

17.6.1.3 Read Length Field

The Read Length byte conveys the number of bytes the target device must supply the originator before returning the Frame Check Sequence (FCS) byte calculated over that data. The Read Length is in the range of 0x00 and 0xFF.

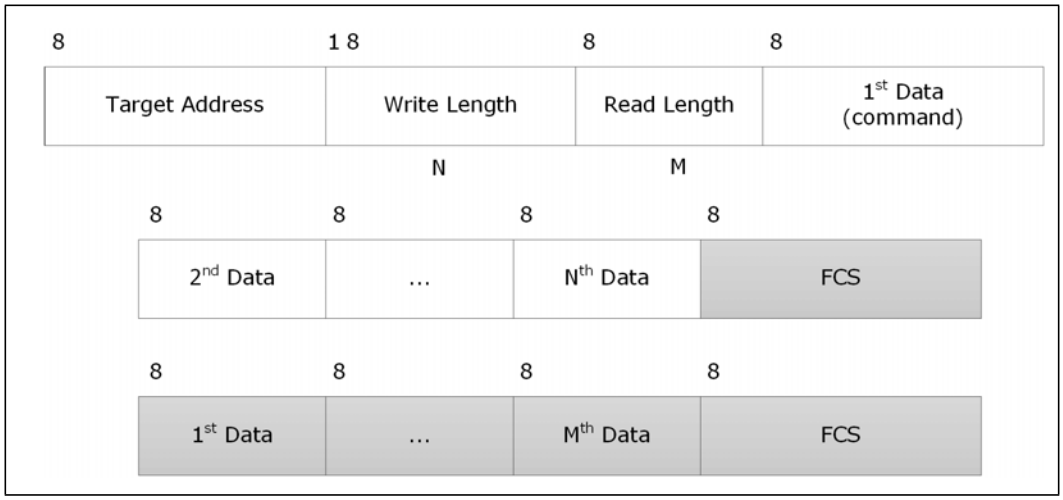
17.6.1.4 Command Byte

All SoC PECI commands except Ping() require a Command byte. The command code is positioned as the first data byte, that is, data_byte[0].

17.6.2 PECI Write-Read Protocol

The write-read protocol shown in Figure 17-7 is the only protocol defined for messaging between devices on the PECI. The write-read protocol allows an atomic operation that first writes and then reads data between an originator and a target. Format of a write-read message is given in Figure 17-7. The read and write Frame Check Sequence (FCS) bytes are not calculated but may be required to have a byte allocated as part of the specific command where described in Section 17.6.2.1, "PECI Proxy Command Format" on page 385.

Figure 17-6. PECI Write-Read Protocol



Legend:

Master to Slave

Slave to Master

Similar to Figure 17-5, "PECI Message Header in the SMBus Packet" on page 382, the PECI Message Header, the Address Timing Negotiation (NT) and Message Timing Negotiation (MT) bits described in *RS - Platform Environment Control Interface (PECI) Specification,* Revision 3.0 are not implemented. It is the responsibility of the BMC to not include these timing bits in the header.

17.6.2.1 PECI Proxy Command Format

The SoC has no Manageability Engine (ME). Unlike some Intel Peripheral Controller Hub (PCH) products, the SoC PECI-over-SMBus controller only operates in PECI mode. The SMBus Interface PECI Proxy Command Protocol is an encapsulation of PECI specification commands onto the SMBus by the SMBus master. Two new SoC-specific PECI commands are also supported.

The PECI proxy command bytes, encapsulated in bytes [N:2] of the protocol, conform to Figure 17-6, "PECI Write-Read Protocol" on page 384 and the *RS - Platform Environment Control Interface (PECI) Specification,* Revision 3.0. The SoC PECI-over-SMBus controller provides a transparent pass-through of the PECI command to the integrated PECI client and does not verify the validity of the command. It is the burden of the BMC or platform board External Circuitry (EC) to provide valid data in the PECI Proxy Command fields of the incoming packet.

Because the SMBus is used and not the serial PECI electrical interface, it is unnecessary to calculate the Frame Check Sequence (FCS). The SoC design does keep the PECI commands consist with non-SoC Intel products and carry additional bytes that are set to 0x0. These are called out where appropriate. The PECI Proxy Command Format is shown in Table 17-5.

Table 17-5. PECI Proxy Command Protocol Format

PECI Proxy Command							
SMBus Default Data			SMBus to PECI Handshake Control Data	PECI Command Data as Described for Each Supported PECI Command in the PECI Specification			
Slave Address	PECI Mode Command Code	Byte Count	Byte 1	Byte 2	Byte 3	Byte 4	Byte [N:5]
0x4C (default)	0x62	0x04...N	Control Byte (Reserved)	PECI Target Address[1] 0x30 = CPU	PECI Write Length[2]	PECI Read Length[3]	Remaining Part of PECI Command[4]

1. Required PECI Header byte.
2. Required PECI Header byte. Must be set to the proper value as defined by the particular PECI Command. If an Assured Write FCS (AW FCS) is needed, the PECI Write Length value must also include the AW FCS byte.
3. Required PECI Header byte. Must be set to the proper value as defined by the particular PECI Command.
4. This field does not exist for the PECI Ping() command. This field is used for write data bytes including a placeholder for an AW FCS byte. Note that the Retry bit shall be set to zero and the command code byte must be one of the codes supported by the SoC.

17.6.2.2 PECI Proxy Read Command

The SMBus Read command uses one SMBus command code (0x40). The SMBus Read command returns PECI Command Status or PECI Response Payload Data from the executed PECI proxy command. Table 17-6 shows the structure of PECI Response Data from a PECI mode Read.

Table 17-6. PECI Proxy Read (Sheet 1 of 2)

SMBus Field	Value	Data Source	Description
Slave Address	0x4B	BMC	Slave address for the Command Phase
Command Code – SMBus Read	0x40	BMC	Command code for SMBus read
Slave Address	0x4B	BMC	Slave address for data phase

Table 17-6. PECI Proxy Read (Sheet 2 of 2)

SMBus Field	Value	Data Source	Description
Byte Count (N)	0x01...N	SoC	N = 0x01 for Busy State: • Byte 1 = 0x01 (CMD_BUSY bit set) N = 0x02 for Transaction Errors • Byte 1 = 0x02 (CMD_ERR bit set) • Byte 2 = Error code N > 2 indicates the request PECI command executed successfully: • Byte 1 = 0x0 • Byte 2 = 0x0 • Byte [N:3] = PECI response data
Byte 1	Control Data		**Status Byte (CMD_STAT):** Bits [7:2] = Reserved Bit 1 = CMD_ERR • This bit is set by the SoC if the PECI proxy interface detects Transaction Error. The Error Code is defined in Byte 2. Bit 0 = CMD_BUSY • This bit is set by the SoC while a PECI command is executing. It is reset by the SoC when the PECI command operation is completed.
Byte 2	Control Data		**Error Code (ERR_CODE):** 0x00 = PECI_ST_OK • PECI command completed successfully 0x01 = PECI_ST_BAD_FORMAT • SMBus PECI message incorrectly formatted 0x09 = PECI_ST_DISABLED • PECI driver disabled 0x3E = PECI_INVALID_SOCKET • Invalid PECI socket number 0xE1 = PECI_ST_TRANS_TIMEOUT_ERROR • PECI Transaction Time-out (e.g., PECI Controller not responding to commands or hangs) 0xFE = PECI_ST_FAILURE • General PECI Failure 0xFF = PECI_ST_UNKNOWN • Unknown Status **Unsupported Error Codes:** 0x02 = PECI_ST_FCS_REQ_ERROR • Error in PECI Request FCS 0x04 = PECI_ST_FCS_RSP_ERROR • Error in PECI Response FCS 0x08 = PECI_ST_LINK_ERROR • PECI Link Error
Byte [N:3]	PECI Response Data		**PECI Response Data:** PECI Response Data as described for each PECI command in the PECI Specification. It contains all data returned by the PECI client that resides between the Write FCS byte and Read FCS byte. These bytes are valid only when the PECI command completes successfully and the CMD_ERR bit in the Status Byte (Byte 1; Bit[1]) is cleared.

17.6.3 PECI Proxy Command Handling Procedure

A BMC requests read and write PECI proxy commands over the SMBus that are formatted to conform to the PECI command protocol. The PECI command handler performs the following on receipt of the SMBus packet:

- Validates that the PECI command is valid.
- Either processes the command locally or forwards the command to other SoC internal units for processing.
- Repackages the command into the appropriate format upon command completion.

The BMC retrieves the PECI response data by performing the SMBus Block Read transactions. The number of data bytes returned from the Block Read command on the SMBus indicates the completion status of the PECI command.

1. If N=1, then only the Status byte is sent with CMB_BUSY bit set (CMD_BUSY=1b).
   Indicates that the PECI command transaction is still in progress.
2. If N=2, then both Status and Error Code bytes are sent with CMD_BUSY bit reset (CMD_BUSY = 0b) and CMD_ERR bit set (CMD_ERR = 1b).
   Indicates that the PECI command resulted in an error. Byte 2 contains the Error Code.
3. If N>2, then both CMD_BUSY and CMD_ERR will be reset (CMD_BUSY = 0b, CMD_ERR = 0b).
   Indicates that the PECI command completed successfully and the read-back data is valid.

Note: The BMC should only trigger new PECI command when the previous command is completed. The SoC does not support posted or multiple commands.

Note: If the Reset PECI SMBus Command is received, any pending state will be reset. The initiator would be expected to reissue the last command.

17.6.4 PECI Proxy Command Trigger

This process provides the functions performed by the SMBus master (the BMC) and the SoC to trigger and complete a PECI Proxy command.

1. BMC performs an SMBus Block write transaction, formatted with the data associated with the requested PECI proxy command as shown in Table 17-6, "PECI Proxy Read" on page 386.
2. The SoC sets the CMD_BUSY bit in the Status byte and handles the requested PECI Command.
3. After completing the PECI operation, the SoC performs the following functions:
   — Resets the CMD_BUSY bit in the Status byte.
   — Sets the CMD_ERR bit if a PECI transaction error occurred; otherwise resets CMD_ERR bit.
   — Sets the ERR_CODE byte with the End of Transaction condition.
   — If the PECI transaction completed successfully, stores the PECI Response Data and transfers it to the BMC on the next Read command.
4. The BMC polls the Status Byte until the CMD_BUSY bit is cleared to indicate the completion of the command. If the PECI transaction completed successfully (CMD_ERR = '0'), the PECI Response data is valid.

17.6.4.1 Unsupported PECI Command

In the case of an unsupported PECI command, the SoC responds with the appropriate error code.

17.6.4.2 Illegally Formatted Command

In the case of an unsupported PECI command, the SoC responds with the appropriate error code.

17.7 PECI Proxy Commands

Table 17-7 is a summary of the PECI Proxy Commands supported by the SoC. Each is described in this section following the table.

Table 17-7. Supported PECI Commands

Command	In PECI Specification, Rev. 3.1	Supported by SoC	Command Code	Definition
Ping()	Yes	Yes	none	This command detects that PECI functionality exists.
GetDIB()	Yes	Yes	0xF7	This command returns the PECI device-specific information.
GetTemp()	Yes	Yes	0x01	This command returns the processor die information.
RdPkgConfig()	Yes	Yes	0xA1	This command reads the SoC package configuration space.
WrPkgConfig()	Yes	Yes	0xA5	This command writes the SoC package configuration space.
RdIAMSR()	Yes	No	-	
WrIAMSR()	Yes	No	-	
RdPCIConfig()	Yes	No	-	
WrPCIConfig()	Yes	No	-	
RdPCIConfigLocal()	Yes	Yes	0xE1	This command reads the SoC PCI space.
WrPCIConfigLocal()	Yes	No	-	
RdEndPointConfig()	No	Yes	0xC1	This command reads the register space over the sideband.
WrEndPointConfig()	No	Yes	0xC5	This command writes the register space over the sideband.

17.7.1 Ping()

Ping() is a required message for all PECI devices. This message is used to enumerate devices or determine if a device has been removed, been powered-off, etc. A Ping() sent to a device address always returns a non-zero Write FCS if the device at the targeted address is able to respond. Table 17-8 shows the Ping PECI Proxy Block Write format and Table 17-9 on page 392 shows the Ping PECI Proxy Block Read format.

Table 17-8. Ping - PECI Proxy Block Write

SMBus	Function		Value	Data Source	Comment
Slave Address (Write)	Write Address	SMBus Command	0x4C	BMC	
Command Code	PECI Mode command code		0x62		
Byte Count	N	Control Data	0x04		
Byte 1	SMBUS-PECI Handshake Control		0x00		
Byte 2	PECI Client Address	PECI Command	0x30		
Byte 3	PECI Write Length		0x00		Ping() command has no command code • Write Length = 0x00 • Read Length = 0x00
Byte 4	PECI Read Length		0x00		

Table 17-9. Ping - PECI Proxy Block Read

SMBus	Function		Value	Data Source	Comment
Slave Address (Command Phase)	Write Address	SMBus Command	0x4B	BMC	
SMBUS Read Command Code	SMBUS Read Code		0x40		
Slave Address (Data Phase)	Read Address		0x4B		
Byte Count	N	Control Data	{0x01, 0x02}	SoC	N = 0x01 for Busy state • Byte 1 = 0x01 N= 0x02 for PECI client status • Byte 1 = 0x00 If Byte 2 = 0x00 • PECI client active If Byte 2 = 0x02 • PECI client not active
Byte 1	Status Byte		CMD_STAT		See Byte[1] of Table 17-6, "PECI Proxy Read" on page 386.
Byte 2	Error Byte		ERR_CODE		If Byte 2 = 0x00 • PECIclient active If Byte 2 = 0x02 • PECI client not active. See Byte[2] of Table 17-6, "PECI Proxy Read" on page 386 for Error Code Definitions.

17.7.2 GetDIB()

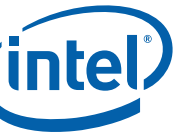
The SoC PECI client implementation of GetDIB() includes an 8-byte response and provides information regarding client revision number and the number of supported domains.Table 17-10 shows the GetDIB PECI Proxy Block Write format and Table 17-11 on page 394 shows the GetDIB PECI Proxy Block Read format.

Table 17-10. GetDIB() PECI Proxy Block Write

SMBus	Function		Value	Data Source	Comment
Slave Address (Write)	Write Address	SMBus Command	0x4C	BMC	
Command Code	PECI Mode command code		0x62		
Byte Count	N	Control Data	0x05		
Byte 1	SMBus-PECI Handshake Control		0x00		Value = 0x00 No AW FCS Required
Byte 2	PECI Client Address	PECI Command	0x30		Socket ID. Always 0x30 for the SoC.
Byte 3	PECI Write Length		0x01		
Byte 4	PECI Read Length		0x08		
Byte 5	GetDIB() Command Code		0xF7		

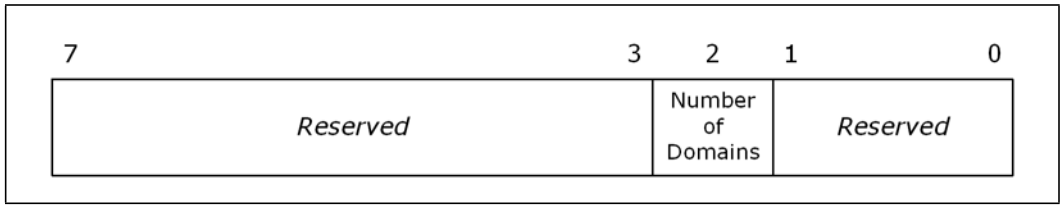
Table 17-11. GetDIB() PECI Proxy Block Read

SMBus	Function		Value	Data Source	Comment
Slave Address (Command Phase)	Write Address	SMBus Command	0x4B	BMC	
SMBus Read Command Code	SMBus Read Code		0x40		
Slave Address (Data Phase)	Read Address		0x4B		
Byte Count	N	Control Data	{0x01, 0x02, 0x0A}	SoC	N = 0x01 for Busy state • Byte 1 = 0x01 N= 0x02 for Transaction Errors • Byte 1 = 0x02 • Byte 2 = Error Code defined in Table 17-6 on page 386. N=0xA indicates successful PECI Transaction • Byte 1 = 0x00 • Byte 2= 0x00 • Byte[10:3] = PECI Response Data
Byte 1	Status Byte		CMD_STAT		See Byte[1] of Table 17-6 on page 386.
Byte 2	Error Byte		ERR_CODE		See Byte[2] of Table 17-6 on page 386.
Byte 3	PECI Response Byte 1	PECI Response Data	PECI Device Info		See Section 17.7.2.1, "PECI Device Info Field" on page 395. Returns the number of Domains in addition to Domain#0 as shown below.
Byte 4	PECI Response Byte 2		PECI Device Revision		See Section 17.7.2.2, "PECI Revision Number" on page 395.
Byte[10:5]	PECI Response Byte [8:3]		Reserved		



17.7.2.1 PECI Device Info Field

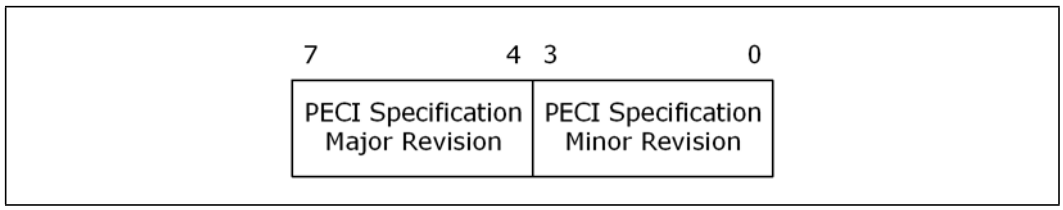
Figure 17-7. PECI Device Info Field Definition



17.7.2.2 PECI Revision Number

The SoC as the client is designed to meet the PECI 3.1 specification. The revision number returned is 0011_0001b.

Figure 17-8. PECI Revision Number Definition



17.7.3 GetTemp()

The GetTemp() command is used to retrieve the temperature from a target PECI address. The temperature is used by the BMC to regulate the temperature on the die. The data is returned as a negative value representing the number of degrees Celsius below the maximum processor Junction Temperature ($T_{J\text{-}MAX}$).

Note that the maximum PECI Temperature value of zero corresponds to the processor $T_{J\text{-}MAX}$. This also represents the default temperature at which the processor Thermal Control Circuit (TCC) activates. The actual value that the thermal management system uses as a control set point ($T_{CONTROL}$) is also defined as a negative number below $T_{J\text{-}MAX}$.

See Section 17.10, "DTS Temperature Data" on page 427 for data format and error codes. Table 17-12 shows the GetTemp PECI Proxy Block Write format and Table 17-13 on page 397 shows the GetTemp PECI Proxy Block Read format.

Table 17-12. GetTemp() PECI Proxy Block Write

SMBus	Function		Value	Data Source	Comment
Slave Address (Write)	Write Address	SMBus Command	0x4C	BMC	
Command Code	PECI Mode command code		0x62		
Byte Count	N	Control Data	0x05		
Byte 1	SMBus-PECI Handshake Control		0x00		Value = 0x00 No AW FCS Required
Byte 2	PECI Client Address	PECI Command	0x30		
Byte 3	PECI Write Length		0x01		
Byte 4	PECI Read Length		0x02		
Byte 5	GetTemp() Command Code		0x01		

Table 17-13. GetTemp() PECI Proxy Block Read

SMBus	Function		Value	Data Source	Comment
Slave Address (Command Phase)	Write Address	SMBus Command	0x4B	BMC	
SMBus Read Command Code	SMBus Read Code		0x40		
Slave Address (Data Phase)	Read Address		0x4B		
Byte Count	N	Control Data	{0x01, 0x02, 0x04}	SoC	N = 0x01 for Busy state • Byte 1 = 0x01 N= 0x02 for Transaction Errors • Byte 1 = 0x02 • Byte 2 = Error Code defined in Table 17-6 on page 386. N=0x4 indicates successful PECI Transaction • Byte 1 = 0x00 • Byte 2= 0x00 • Byte[4:3] = PECI Response Data
Byte 1	Status Byte		CMD_STAT		See Byte[1] of defined in Table 17-6 on page 386.
Byte 2	Error Byte		CMD_ERR		See Byte[2] of defined in Table 17-6 on page 386.
Byte 3	PECI Response Byte 1	PECI Response Data	PECI Device Temp [7:0]		See Section 17.10.1, "PECI Device Temp Data" on page 427.
Byte 4	PECI Response Byte 2		PECI Device Temp [15:8]		

17.7.4 RdPkgConfig()

The RdPkgConfig() command provides read access to the package configuration space (PCS) within the processor including various power and thermal management functions. Typical PCS read services supported by the processor may include access to temperature data, energy status, run time information, DIMM temperatures and so on.

This command does provide multi-domain support. Table 17-14 shows the RdPkgConfig PECI Proxy Block Write format and Table 17-15 on page 399 shows the RdPkgConfig PECI Proxy Block Read format.

Table 17-14. RdPkgConfig() PECI Proxy Block Write

SMBus	Function		Value	Data Source	Comment
Slave Address (Write)	Write Address	SMBus Command	0x4C	BMC	
Command Code	PECI Mode command code		0x62		
Byte Count	N	Control Data	0x09		
Byte 1	SMBus-PECI Handshake Control		0x00		Value = 0x00 No AW FCS Required
Byte 2	PECI Client Address	PECI Command	0x30		
Byte 3	PECI Write Length		0x05		
Byte 4	PECI Read Length		{0x02, 0x3, 0x5}		
Byte 5	RdPkgConfig() Command Code		0xA1		
Byte 6	Host ID and Retry		0x00		
Byte 7	Index				See Section 17.8, "DRAM Thermal Capabilities" on page 410 and Section 17.9, "CPU Thermal and Power Optimization Capabilities" on page 412 for the supported capabilities and their Index and Parameter information.
Byte 8	Parameter LSB				
Byte 9	Parameter MSB				

Table 17-15. RdPkgConfig() PECI Proxy Block Read

SMBus	Function		Value	Data Source	Comment
Slave Address (Command Phase)	Write Address	SMBus Command	0x4B	BMC	
SMBus Read Command Code	SMBus Read Code		0x40		
Slave Address (Data Phase)	Read Address		0x4B		
Byte Count	N	Control Data	{0x01, 0x02, 0x04, 0x05, 0x07}	SoC	N = 0x01 for Busy state • Byte 1 = 0x01 N= 0x02 for Transaction Errors • Byte 1 = 0x02 • Byte 2 =Error Code defined in Table 17-6 on page 386. N={0x4,0x5,0x7} indicates successful PECI Transaction • Byte 1 = 0x00 • Byte 2= 0x00 • Byte[7:3] = PECI Response Data N = {0x04, 0x05, 0x07} value is determined by the number of PECI Response Data bytes returned in Byte[7:4] on the SMBus. N Value: • 0x04 = 1 data byte • 0x05 = 2 data bytes • 0x07 = 4 data bytes
Byte 1	Status Byte		CMD_STAT		See Byte[1] of defined in Table 17-6 on page 386.
Byte 2	Error Byte		ERR_CODE		See Byte[2] of defined in Table 17-6 on page 386.
Byte 3	PECI Transaction Status	PECI Response Data	PECI Completion Code		Completion Code decode: • 0x40 = Command successful • 0x80 = Response time-out • 0x90 = Illegal command
Byte 4	Data 1 (LSB)		Data = 1, 2, or 4 bytes		See Section 17.8, "DRAM Thermal Capabilities" on page 410 and Section 17.9, "CPU Thermal and Power Optimization Capabilities" on page 412 for the supported capabilities and their returned Data Structures.
Byte 5	Data 2				
Byte 6	Data 3				
Byte 7	Data 4 (MSB)				

17.7.5 WrPkgConfig()

The WrPkgConfig() command provides write access to the Package Configuration Space (PCS) within the processor including various power and thermal management functions. Typical PCS write services supported by the processor may include power limiting, thermal averaging constant programming, and other write services. Table 17-16 shows the WrPkgConfig PECI Proxy Block Write format and Table 17-17 on page 402 shows the WrPkgConfig PECI Proxy Block Read format.

Table 17-16. WrPkgConfig() PECI Proxy Block Write

SMBus	Function		Value	Data Source	Comment
Slave Address (Write)	Write Address	SMBus Command	0x4C	BMC	
Command Code	PECI Mode command code		0x62		
Byte Count	N	Control Data	{0x0B, 0x0C, 0x0E}		N = {0x0b, 0x0c, 0x0e} value is determined by the number of PECI Write Data bytes transmitted in Byte[13:10] on the SMBus. N Value: • 0x0b = 1 data byte • 0x0c = 2 data bytes • 0x0e = 4 data bytes
Byte 1	SMBus-PECI Handshake Control		0x01		Value = 0x01 • Assured Write (AW) Frame-Check Sequence (FCS) is not required. However, to maintain compatibility included is the AW space for the byte per PECI specification.
Byte 2	PECI Client Address	PECI Command	0x30		
Byte 3	PECI Write Length		{0x07, 0x08, 0x0A}		Refer to later Section
Byte 4	PECI Read Length		0x01		
Byte 5	WrPkgConfig() Command Code		0xA5		
Byte 6	Host ID and Retry		0x00		
Byte 7	Index				See Section 17.8, "DRAM Thermal Capabilities" on page 410 and Section 17.9, "CPU Thermal and Power Optimization Capabilities" on page 412 for the supported capabilities and their Index, Parameter, and Data Structure information.
Byte 8	Parameter (LSB)				
Byte 9	Parameter (MSB)				
Byte 10	PECI Data 1 (LSB)				
Byte 11	PECI Data 2				
Byte 12	PECI Data 3				
Byte 13	PECI Data 4 (MSB)				
Byte 14	AW FCS Byte		0x00		Dummy to maintain compatibility with PECI specification.

Table 17-17. WrPkgConfig() PECI Proxy Block Read

SMBus	Function		Value	Data Source	Comment
Slave Address (Command Phase)	Write Address	SMBus Command	0x4B	BMC	
SMBus Read Command Code	SMBus Read Code		0x40		
Slave Address (Data Phase)	Read Address		0x4B		
Byte Count	N	Control Data	{0x01, 0x02, 0x03}	SoC	N = 0x01 for Busy state • Byte 1 = 0x01 N= 0x02 for Transaction Errors • Byte 1 = 0x02 • Byte 2 =Error Code defined in Table 17-6 on page 386. N={0x03} indicates successful PECI Transaction • Byte 1 = 0x00 • Byte 2= 0x00 • Byte3 = PECI Completion Code
Byte 1	Status Byte		CMD_STAT		See Byte[1] of defined in Table 17-6 on page 386.
Byte 2	Error Byte		ERR_CODE		See Byte[2] of defined in Table 17-6 on page 386.
Byte 3	PECI Transaction Status	PECI Response Data	PECI Completion Code		Completion Code decode: • 0x40 = Command successful • 0x80 = Response time-out • 0x90 = Illegal command

17.7.6 RdPCIConfigLocal()

The RdPCIConfigLocal() command provides sideband read access to the entire PCI configuration space of the SoC. This command allows access to all of the configuration space that can also be accessed in-band using the Enhanced Configuration Access Mechanism (ECAM). Using the PECI access mechanism, PECI originators can access the SoC configuration space even before the BIOS enumeration of the system PCI busses. PECI originators may also conduct a device/function/register enumeration sweep of this space by issuing reads in the same manner that the BIOS would. Table 17-18 shows the RdPCIConfigLocal PECI Proxy Block Write format and Table 17-19 on page 404 shows the RdPCIConfigLocal PECI Proxy Block Read format. It is not possible to access PCI Express* B0,D0,F0 using the RdPCIConfigLocal command.

Table 17-18. RdPCIConfigLocal() PECI Proxy Block Write

SMBus	Function		Value	Data Source	Comment
Slave Address (Write)	Write Address	SMBus Command	0x4C	BMC	
Command Code	PECI Mode Command Code		0x62		
Byte Count	N	Control Data	0x09		
Byte 1	SMBus-PECI Handshake Control		0x00		Value = 0x00 No AW FCS Required
Byte 2	PECI Client Address	PECI Command	0x30		
Byte 3	PECI Write Length		0x05		
Byte 4	PECI Read Length		{0x02,0x3, 0x5}		One byte for the Completion Code plus byte count of the desired data return from the accessed Configuration-Space Register: 0x02 = One Byte desired. 0x03 = One Word desired. 0x05 = One DWord desired.
Byte 5	RdIPCILocal() Command Code		0xe1		
Byte 6	Host ID and Retry		0x00		
Byte 7	PCI Config Address Byte 1 (LSB)		24-bit PCI Configuration Address		24-Bit PCI Configuration Address Mapping: • Bit[11:0] = Register • Bit[14:12] = Function • Bit[19:15] = Device • Bit[23:20] = Bus
Byte 8	PCI Config Address Byte 2				
Byte 9	PCI Config Address Byte 3 (MSB)				

Table 17-19. RdPCIConfigLocal() PECI Proxy Block Read

SMBus	Function		Value	Data Source	Comment
Slave Address (Command Phase)	Write Address	SMBus Command	0x4B	BMC	
SMBus Read Command Code	SMBus Read Code		0x40		
Slave Address (Data Phase)	Read Address		0x4B		
Byte Count	N	Control Data	{0x01, 0x02, 0x04, 0x05, 0x07}	SoC	N = 0x01 for Busy state • Byte 1 = 0x01 N = 0x02 for Transaction Errors • Byte 1 = 0x02 • Byte 2 = Error Code defined in Table 17-6 on page 386. N = {0x4,0x5,0x7} indicates successful PECI Transaction • Byte 1 = 0x00 • Byte 2 = 0x00 • Byte[7:3] = PECI Response Data N = {0x04, 0x05, 0x07} value is determined by the number of PECI Response Data bytes returned in Byte[7:4] on the SMBus. N Value: • 0x04 = 1 data byte • 0x05 = 2 data bytes • 0x07 = 4 data bytes
Byte 1	Status Byte		CMD_STAT		See Byte[1] of defined in Table 17-6 on page 386.
Byte 2	Error Byte		ERR_CODE		See Byte[2] of defined in Table 17-6 on page 386.
Byte 3	PECI Transaction Status	PECI Response Data	PECI Completion Code		Completion Code decode: • 0x40 = Command successful • 0x80 = Response time-out • 0x90 = Illegal command
Byte 4	PCI Data 1 (LSB)		Data = 1,2 or 4 bytes		Data returned from PCI Configuration Space. Data is either a Byte, Word, or DWord depending on the PECI Read Length field of the associated PECI Proxy Block Write.
Byte 5	PCI Data 2				
Byte 6	PCI Data 3				
Byte 7	PCI Data 4 (MSB)				

17.7.7 RdEndPointConfig()

The RdEndPointConfig() command provides sideband read access to the PCI configuration space that resides within the processor, as well as the SSA sideband configuration space of each agent. The exact listing of supported devices, functions and registers is outside the scope of this document. PECI originators can access this space even before BIOS enumeration of the system busses. PECI originators may also conduct a device/function/register enumeration sweep of this space by issuing reads in the same manner that the BIOS would. Table 17-20 shows the RdEndPointConfig PECI Proxy Block Write format and Table 17-21 on page 406 shows the RdEndPointConfig PECI Proxy Block Read format.

The following Endpoints are accessible BUNIT (0x3), PUNIT (0x4), DUNIT0 (0x10), DUNIT1 (0x13), TUNIT (0x2).

Table 17-20. RdEndPointConfig() PECI Proxy Block Write

SMBus	Function		Value	Data Source	Comment
Slave Address (Write)	Write Address	SMBus Command	0x4C	BMC	
Command Code	PECI Mode command code		0x62		
Byte Count	N	Control Data	0x0B		
Byte 1	SMBus-PECI Handshake Control		0x00		Value = 0x00 No AW FCS Required
Byte 2	PECI Client Address	PECI Command	0x30		
Byte 3	PECI Write Length		{0x07}		
Byte 4	PECI Read Length		{0x02, 0x3, 0x5}		
Byte 5	RdEndPointConfig Command Code		0xC1		Use the same command for both, i.e., RdEndPointConfig is an alias.
Byte 6	Host ID and Retry		0x00		
Byte 7	SoC Sideband Port		0x02 0x03 0x04 0x10 0x13		Sideband Port • 0x02 = T-Unit • 0x03 = B-Unit • 0x04 = P-Unit • 0x10 = D-Unit0 • 0x13 = D-Unit1 All other values return an Error.
Byte 8	Port Register Address Byte 1 (LSB)		Any valid register address for the sideband port		Port offset If the addressed sideband port register does not exist, an Error is returned.
Byte 9	Port Register Address Byte 2				
Byte 10	Port Register Address Byte 3				
Byte 11	Port Register Address Byte 4 (MSB)				

Table 17-21. RdEndPointConfig() PECI Proxy Block Read

SMBus	Function		Value	Data Source	Comment
Slave Address (Command Phase)	Write Address	SMBus Command	0x4B	BMC	
SMBus Read Command Code	SMBus Read Code		0x40		
Slave Address (Data Phase)	Read Address		0x4B		
Byte Count	N	Control Data	{0x1, 0x02, 0x04, 0x05, 0x07}	SoC	N = 0x01 for Busy state • Byte 1 = 0x01 N= 0x02 for Transaction Errors • Byte 1 = 0x02 • Byte 2 =Error Code defined in Table 17-6 on page 386. N={0x4,0x5,0x7} indicates successful PECI Transaction • Byte 1 = 0x00 • Byte 2= 0x00 • Byte[7:3] = PECI Response Data N = {0x04, 0x05, 0x07} value is determined by the number of PECI Response Data bytes returned in Byte[7:4] on the SMBus. N Value: • 0x04 = 1 data byte • 0x05 = 2 data bytes • 0x07 = 4 data bytes
Byte 1	Status Byte		CMD_STAT		See Byte[1] of defined in Table 17-6 on page 386.
Byte 2	Error Byte		ERR_CODE		See Byte[2] of defined in Table 17-6 on page 386.
Byte 3	PECI Transaction Status	PECI Response Data	PECI Completion Code		Completion Code decode: • 0x40 = Command successful • 0x80 = Response time-out • 0x90 = Illegal command
Byte 4	Data 1 (LSB)		Data = 1, 2, or 4 bytes		See later section.
Byte 5	Data 2				
Byte 6	Data 3				
Byte 7	Data 4 (MSB)				

17.7.8 WrEndPointConfig()

The WrEndPointConfig() command provides sideband write access to the PCI configuration space that resides within the processor. PECI originators can access this space even before BIOS enumeration of the system busses. The exact listing of supported devices, functions and registers for writing is outside the scope of this document. Refer to the appropriate processor specifications for details on registers that are accessible through this command. Table 17-22 shows the RdEndPointConfig PECI Proxy Block Write format and Table 17-23 on page 409 shows the RdEndPointConfig PECI Proxy Block Read format.

WrEndPointConfig only supports Endpoint agent Punit (0x4). The registers that can be modified are the thermal registers in the following address ranges:

0x80 - 0x8E
0xB2
0xB4 - 0xC2

Table 17-22. WrEndPointConfig() PECI Proxy Block Write

SMBus	Function		Value	Data Source	Comment
Slave Address (Write)	Write Address	SMBus Command	0x4C	BMC	
Command Code	PECI Mode Command Code		0x62		
Byte Count	N	Control Data	{0x0D, 0x0E, 0x10}		
Byte 1	SMBus-PECI Handshake Control		0x01		Always set the value of Byte 1 to 0x01.
Byte 2	PECI Client Address	PECI Command	0x30		
Byte 3	PECI Write Length		{0x09, 0x0A, 0x0C}		Refer to later section.
Byte 4	PECI Read Length		0x01		
Byte 5	WrEndPointConfig() Command Code		0xC5		
Byte 6	Host ID & Retry		0x00		
Byte 7	SoC Sideband Port		0x04		Sideband Port • 0x04 = P-Unit All other values return an error.
Byte 8	Register Byte 1 (LSB)		Address		Port 04 offset. Valid offset values are: • 0x0080 through 0x008E • 0x00B2 0x00B4 through 0x00C2
Byte 9	Register Byte 2				
Byte 10	Register Byte 3				
Byte 11	Register Byte 4 (MSB)				
Byte 12	PECI Data 1 (LSB)		Data = 1, 2, or 4 bytes		Sidebaoperation.
Byte 13	PECI Data 2				
Byte 14	PECI Data 3				
Byte 15	PECI Data 4 (MSB)				
Byte 16	AW FCS		0x00		Always set the value of Byte 16 to 0x00.

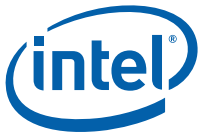


Table 17-23. WrEndPointConfig() PECI Proxy Block Read

SMBus	Function		Value	Data Source	Comment
Slave Address (Command Phase)	Write Address	SMBus Command	0x4B	BMC	
SMBus Read Command Code	SMBus Read Code		0x40		
Slave Address (Data Phase)	Read Address		0x4B		
Byte Count	N	Control Data	{0x03}	SoC	N = 0x01 for Busy state • Byte 1 = 0x01 N = 0x02 for Transaction Errors • Byte 1 = 0x02 • Byte 2 =Error Code defined in Table 17-6 on page 386. N = {0x03} indicates successful PECI Transaction • Byte 1 = 0x00 • Byte 2 = 0x00 • Byte3 = PECI Completion Code
Byte 1	Status Byte		CMD_STAT		See Byte[1] defined in Table 17-6 on page 386.
Byte 2	Error Byte		ERR_CODE		See Byte[2] defined in Table 17-6 on page 386.
Byte 3	PECI Transaction Status	PECI Response Data	PECI Completion Code		Completion Code decode: • 0x40 = Command successful • 0x80 = Response time-out • 0x90 = Illegal command

17.8 DRAM Thermal Capabilities

Various DRAM component temperature data can be accessed using the RdPkgConfig() and WrPkgConfig() PECI commands addressed to the SoC. See Section 17.7.4, "RdPkgConfig()" on page 398 and Section 17.7.5, "WrPkgConfig()" on page 400 for the format of the commands. The accessible data structures are given in Table 17-24 which contains the values of the command's Index, Parameter, and PECI Data fields.

Table 17-24. Summary of DRAM Thermal Services

Service	Index Field (Decimal)	Parameter Field (Word)	PECI Data Field (DWord)	Description	Alternate In-band MSR or CSR Access	Supported
WrPkgConfig() DRAM Rank Temperature Write	18	Channel Index and DIMM Index[1]	Absolute temperature in Degrees Celsius for ranks 0-3[2]	Write the temperature for each Rank within a single DIMM	none	Yes Section 17.8.1
RdPkgConfig() DRAM Channel Temperature Read	22	0x0000	Maximum of all rank temperatures for each channel in Degrees Celsius	Read the maximum DRAM Channel Temperature	none	Yes Section 17.8.2
WrPkgConfig() RdPkgConfig() DIMM Ambient Temperature Write/Read	19	0x0000	Absolute temperature in Degrees C to be used as ambient temperature reference	Write/Read ambient temperature reference for activity-based Rank Temperature estimation	none	No

1. See Figure 17-9 and Table 17-25.
2. See Figure 17-11 on page 411.

Figure 17-9. Channel Index and DIMM Index Parameter Word

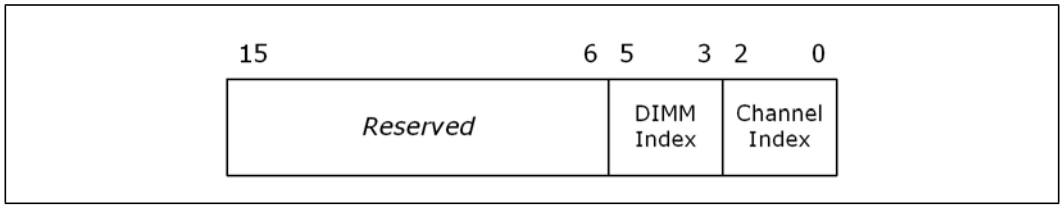


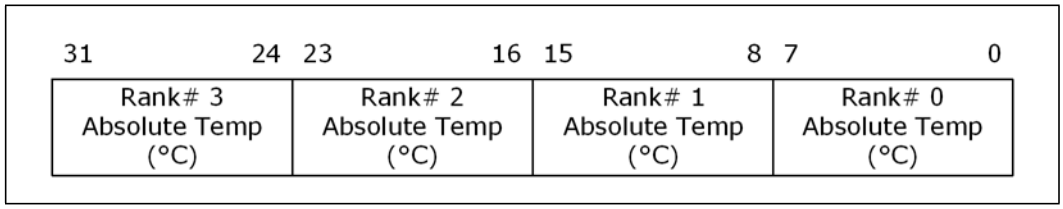
Table 17-25. Channel Index and DIMM Index

Index Encoding	Physical Channel Number	Physical DIMM Number
000	0	0
001	1	1
010	Reserved	2
011	Reserved	3

17.8.1 DRAM Rank Temperature Write (Index = 18)

See Table 17-24, "Summary of DRAM Thermal Services" on page 410. The DRAM Rank Temperature Write allows the PECI host to program the processor with the temperature for all the ranks within a DIMM up to a maximum of four ranks. The programming data structure is defined in Figure 17-10.

Figure 17-10.Write DRAM Rank Temperature Data DWord



The DIMM index and Channel index are specified through the Parameter field as shown in Figure 17-9 on page 410 and Table 17-25 on page 410. This write is relevant to platforms that do not have on-die or on-board DIMM thermal sensors to provide memory temperature information or if the processor does not have direct access to the DIMM thermal sensors. This temperature information is used by the processor in conjunction with the activity-based DRAM temperature estimations. With this SoC product family, while temperature values are accepted on a per-rank basis, the SoC uses the average temperature over a given channel for throttling decisions.

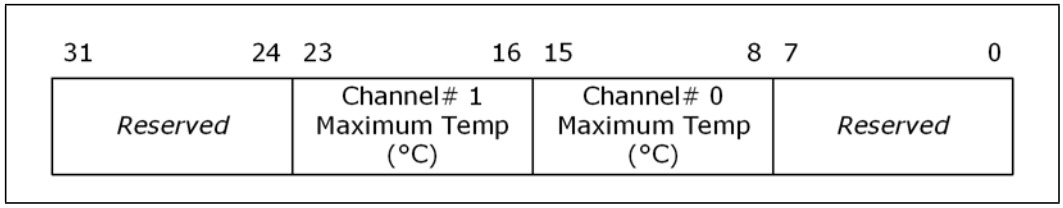
17.8.2 DRAM Channel Temperature Read (Index = 22)

See Table 17-24, "Summary of DRAM Thermal Services" on page 410. This feature enables a PECI host read of the maximum temperature of each Memory Controller Channel of the given product SKU. This includes all of the DIMMs within the Channel and all the Ranks within each of the DIMMs. See the returned data structure as shown in Figure 17-11 on page 411.

Note: Memory Channels not populated by DRAM return:

- the Ambient Temperature on systems using activity-based temperature estimations,
  or alternatively,
- a zero for systems using sensor-based temperatures.

Figure 17-11.Read DRAM Channel Temperature Data DWord



17.9 CPU Thermal and Power Optimization Capabilities

Table 17-26 provides a summary of the power and thermal optimization capabilities that can be accessed over PECI for this SoC product family. Twenty-three services are shown and each is described in the subsections following the table. Note that the Index Field values are referenced as decimal numbers.

Table 17-26 also shows alternate in-band mechanisms to access similar or equivalent information where applicable for register read and write services.

The BIOS is required to populate CPUID, PlatformID and CPU Microcode Update Revision. See section 23.5.1 - 23.5.3 of the *Intel® Atom™ Processor C2000 Product Family - BIOS Writer's Guide (BWG)*, Volume 2.

Table 17-26. Summary of CPU Thermal and Power Optimization Services (Sheet 1 of 4)

Service	Index Field (Decimal)	Parameter Field (Word)	PECI Data Field (DWord)	Description	Alternate In-band MSR or CSR Access
RdPkgConfig() Package Identifier Read Section 17.9.1, on page 416	0	0x0000	CPU ID Information	Returns processor-specific information including CPU family, model and stepping information.	Execute the CPUID instruction to get the processor signature
		0x0001	Platform ID	Used to ensure microcode update compatibility with processor.	IA32_PLATFORM_ID (MSR)
		0x0003	Max Thread ID	Returns the maximum Thread ID value supported by the processor.	RESOLVED_CORES_MASK (MSR & CSR)
		0x0004	CPU Microcode Update Revision	Returns processor microcode and internal power control-unit firmware revision information.	IA32_BIOS_SIGN_ID (MSR)
		0x0005	MCA Error Source Log	Returns the MCA Error Source Log	MCA_ERR_SRC_LOG (CSR)
RdPkgConfig() Package Temperature Read Section 17.9.5, on page 420	2	0x00FF	Processor package Temperature	Returns the maximum processor die temperature in PECI format. To get the equivalent of the architectural MSR IA_PACKAGE_THERM_STATUS, read IA32_CR_THERM_STATUS for each core and take the maximum value that was read.	IA32_CR_THERM_STATUS
RdPkgConfig() Accumulated Energy Status Read Section 17.9.11, on page 423	3	0x00FF: CPU package	Accumulated CPU energy	Returns the value of the energy consumed by entire SoC.	ENERGY_ STATUS (MSR) PACKAGE_ENERGY_STATUS (CSR)

Table 17-26. Summary of CPU Thermal and Power Optimization Services (Sheet 2 of 4)

Service	Index Field (Decimal)	Parameter Field (Word)	PECI Data Field (DWord)	Description	Alternate In-band MSR or CSR Access
WrPkgConfig() "Wake on PECI" Mode bit Write Section 17.9.14, on page 426	5	0x0001: Set 0x0000: Reset	"Wake on PECI" mode bit	Enables waking-up of SoC from a lower Package State.	none
RdPkgConfig() "Wake on PECI" Mode bit Read Section 17.9.14, on page 426	5	0x0000	"Wake on PECI" mode bit	Read status of "Wake on PECI" mode bit.	none
RdPkgConfig() Package Power Limit Performance Status Read Section 17.9.13, on page 426	8	0x00FF: CPU package	Accumulated CPU throttle time	Read the total time for which the processor package was throttled due to power limiting.	PACKAGE_RAPL_PERF_STATUS (CSR)
RdPkgConfig() Per Core DTS Temperature Read Section 17.9.6, on page 420	9	0x0000 *through* 0x0007: Cores 0 *through* 7 0x00FF: System Agent + C20	Per core DTS maximum temperature	Read the maximum DTS temperature of a particular core or the System Agent within the processor die in relative PECI temperature format.	IA32_CR_THERM_STATUS To get the equivalent of the architectural MSR IA_PACKAGE_THERM_STATUS, read IA32_CR_THERM_STATUS for each core and take the maximum value that was read.
RdPkgConfig() Temperature Target Read Section 17.9.7, on page 421	16	0x0000	Processor $T_{J\text{-}MAX}$ and $T_{CONTROL}$	Returns the maximum processor junction temperature and processor $T_{CONTROL}$.	IA32_TEMPERATURE_TARGET (MSR) TEMPERATURE_TARGET (CSR)
RdPkgConfig() Current Limit Read Section 17.9.10, on page 422	17	0x0000	Current Limit per power plane	Reads the current limit on the VCC power plane in 1/8 of an ampere.	none
RdPkgConfig() Thermal Averaging Constant Read Section 17.9.8, on page 421	21	0x0000	Thermal Averaging Constant	Reads the Thermal Averaging Constant.	none

Table 17-26. Summary of CPU Thermal and Power Optimization Services (Sheet 3 of 4)

Service	Index Field (Decimal)	Parameter Field (Word)	PECI Data Field (DWord)	Description	Alternate In-band MSR or CSR Access
WrPkgConfig() Thermal Averaging Constant Write Section 17.9.8, on page 421	21	0x0000	Thermal Averaging Constant	Writes the Thermal Averaging Constant.	none
WrPkgConfig() Package Power limits For multiple Turbo Modes Section 17.9.12, on page 424	26	0x0000	Power Limit 1 Data	Write Power Limit 1 Data in multiple turbo mode.	TURBO_POWER_LIMIT (CSR) PKG_POWER_LIMIT (MSR)
RdPkgConfig() Package Power limits For multiple Turbo Modes Section 17.9.12, on page 424	26	0x0000	Power Limit 1 Data	Read Power Limit 1 Data in multiple turbo mode.	TURBO_POWER_LIMIT (CSR) PKG_POWER_LIMIT (MSR)
WdPkgConfig() Package Power limits For multiple Turbo Modes Section 17.9.12, on page 424	27	0x0000	Power Limit 2 Data	Write Power Limit 2 Data in multiple turbo mode.	TURBO_POWER_LIMIT (CSR) PKG_POWER_LIMIT (MSR)
RdPkgConfig() Package Power Limits For multiple Turbo Modes Section 17.9.12, on page 424	27	0x0000	Power Limit 2 Data	Read Power Limit 2 Data in multiple turbo mode.	TURBO_POWER_LIMIT (CSR) PKG_POWER_LIMIT (MSR)
RdPkgConfig() Package Power SKU Read Section 17.9.3, on page 419	28	0x0000	Package Power SKU [31:0]	Returns Thermal Design Power (TDP) and minimum package power for the SoC product SKU.	PACKAGE_POWER_SKU (CSR)
RdPkgConfig() Package Power SKU Read Section 17.9.3, on page 419	29	0x0000	Package Power SKU[64:32]	Returns the maximum package power value for the SoC product SKU and the maximum time interval for which it can be sustained.	PACKAGE_POWER_SKU (CSR)

Table 17-26. Summary of CPU Thermal and Power Optimization Services (Sheet 4 of 4)

Service	Index Field (Decimal)	Parameter Field (Word)	PECI Data Field (DWord)	Description	Alternate In-band MSR or CSR Access
RdPkgConfig() Package Power SKU Unit Read Section 17.9.2, on page 418	30	0x0000	Time, Energy and Power Units	Read units for power, energy and time used in power control registers.	PACKAGE_POWER_SKU_UNIT (MSR and CSR)
RdPkgConfig() Accumulated Run Time Read Section 17.9.4, on page 420	31	0x0000	Total reference time	Returns the total run time in ms. It is an approximation and may not match the TSC.	IA32_TIME_STAMP_COUNTER (MSR)
RdPkgConfig() Thermally Constrained Time Read Section 17.9.9, on page 422	32	0x0000	Thermally Constrained Time	Read the time for which the processor has been operating in a lowered power state due to internal TCC activation.	none
RdPkgConfig() SoC Power Budget Section 17.9.15, on page 426	40	0x0000	Returns the Power Budget as defined in sideband register 0x2	SoC-consumed power in POWER_UNIT_FORMAT	SOC_POWER_BUDGET (CSR)

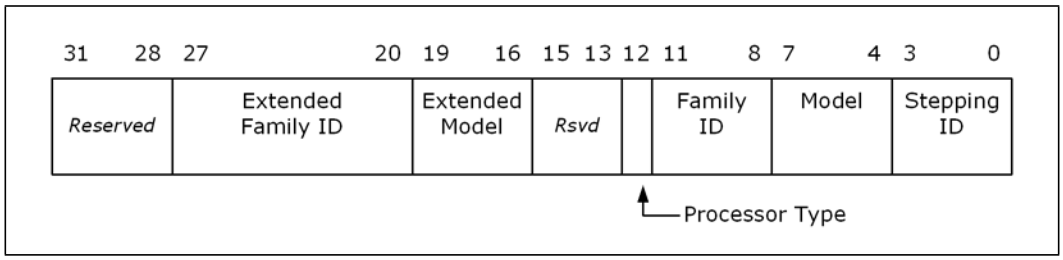
17.9.1 Package Identifier Read (Index = 0)

This feature enables the PECI host to uniquely identify the PECI client processor. The parameter field encodings shown in Table 17-26 on page 412 allow the PECI host, typically the BMC, to access the relevant processor information as described below.

17.9.1.1 CPU ID Information

This is data field contains the equivalent information that can be accessed through executing the Intel® architecture CPUID instruction. It contains the processor type, stepping, model and family ID information as shown in Figure 17-12.

Figure 17-12.CPU ID Data



17.9.1.2 Platform ID

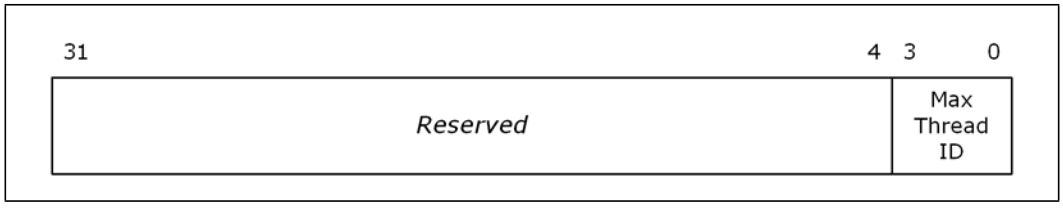
This is data field can be used to ensure processor microcode updates are compatible with the processor. Note that the value of the Platform ID or Processor Flag[2:0] as shown in Figure 17-13 is typically unique to the platform type and processor stepping. Refer to the processor *Intel® Atom™ Processor C2000 Product Family - BIOS Writer's Guide (BWG)* for more information.

Figure 17-13.Platform ID Data

17.9.1.3 Max Thread ID

This is data field provides the number of supported processor threads. Note that this value is dependent on the number of cores within the processor as determined by the processor SKU and is independent of whether certain cores or corresponding threads are enabled or disabled. See Figure 17-14.

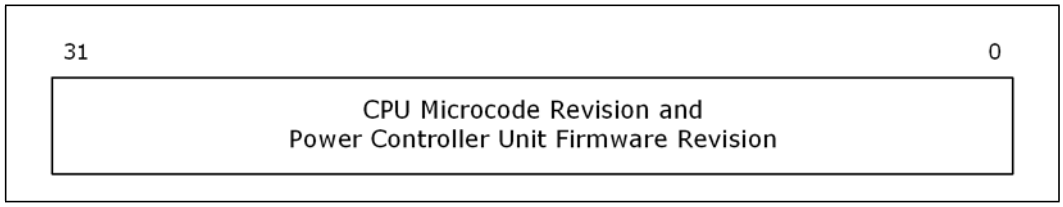
Figure 17-14.Maximum Thread ID Data



17.9.1.4 CPU Microcode Update Revision

This is data field reflects the revision number for the microcode update and power control unit firmware updates on the processor sample. Note that the revision data is a unique 32-bit identifier that reflects a combination of specific versions of the processor microcode and power control unit firmware. See Figure 17-15.

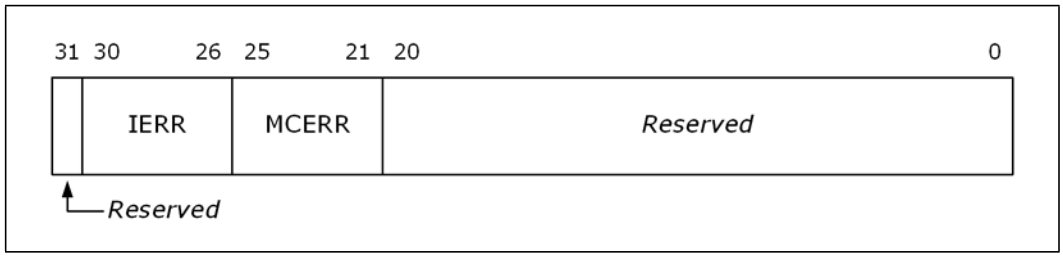
Figure 17-15.Processor Microcode Revision



17.9.1.5 MCA Error Source Log

This is data field contains contents of the Machine-Check Architecture (MCA) Error Source Log register. See Figure 17-16 for details. The register indicates the value as defined when IERR and/or MCERR are indicated by the SoC.

Figure 17-16.Machine Check Status



17.9.2 Package Power SKU Unit Read (Index = 30)

This feature enables the PECI host to read the units of time, energy and power used in the processor and DRAM power control registers for calculating power and timing parameters. In Figure 17-17, the default values are:

- Power Unit field [3:0] = 0011b
- Energy Unit field [12:8] = 10000b
- Time Unit field [19:16] = 1010b

Actual unit values are calculated as shown in Table 17-27.

Figure 17-17. Package Power SKU Unit Data

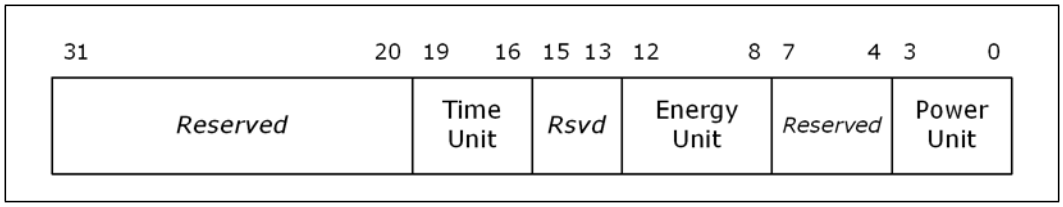


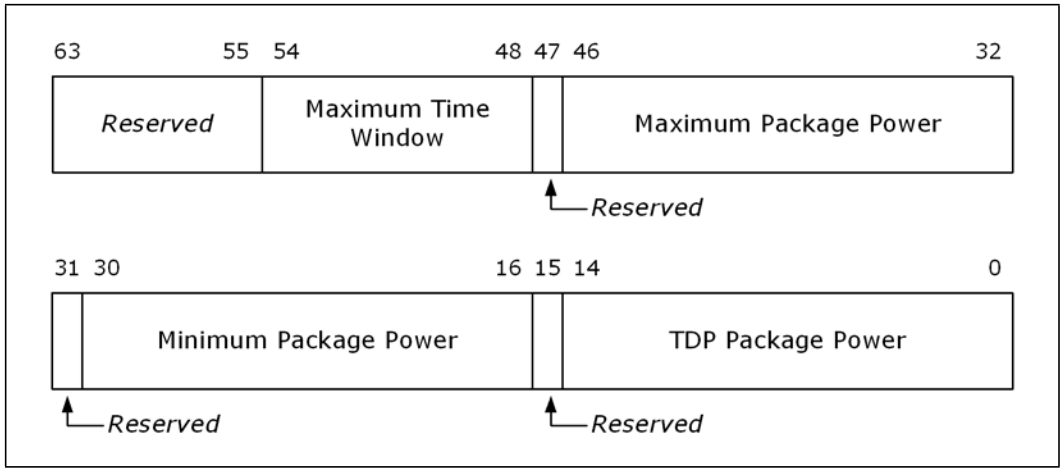
Table 17-27. Power Control Register Unit Calculations

Unit Field	Value Calculation	Default Value
Time	1s / 2^TIME UNIT	1s / 2^10 = 976 µs
Energy	1/2^ENERGY UNIT	1/2^16 = 15.3 µJ
Power	2^POWER UNIT in mW	2^3 * 1mW = 32 mW

17.9.3 Package Power SKU Read (Index = 28 and 29)

This read allows the PECI host to access the minimum, Thermal Design Power (TDP) and maximum power settings for the processor package SKU. It also returns the maximum time interval or window over which the power can be sustained. If the power limiting entity specifies a power limit value outside of the range specified through these settings, power regulation cannot be guaranteed. Since this data is 64 bits wide, PECI facilitates access to this register by allowing two requests to read the lower 32 bits and upper 32 bits separately as shown in Figure 17-18 on page 419. Power units for this read are determined as per the Package Power SKU Unit settings described in Section 17.9.2, "Package Power SKU Unit Read (Index = 30)" on page 418.

Figure 17-18.Package Power SKU Data



The Package Power SKU data is programmed by the SoC internal Power Controller Unit (PCU) firmware during boot time based on SKU dependent power-on default values set during SoC manufacturing. The TDP Package Power specified through bits [14:0] in Figure 17-18 is the maximum value of the Power Limit1 field described in Section 17.9.12, "Package Power Limits For Multiple Turbo Modes (Index = 26 and 27)" on page 424 while the Maximum Package Power in bits [46:32] is the maximum value of the Power Limit2 field which is also described in Section 17.9.12.

The Minimum Package Power in bits [30:16] is applicable to both the Power Limit1 and Power Limit2 fields and corresponds to a mode when all the cores are operational and in their lowest frequency mode. Attempts to program the power limit below the minimum power value may not be effective since BIOS/OS, and not the integrated PCU, controls disabling of cores and core activity.

Maximum Time Window' in bits [54:48] is representative of the maximum rate at which the internal PCU can sample the package energy consumption and reactively take the necessary measures to meet the imposed power limits. Programming too-large of a time window runs the risk of the PCU not being able to monitor and take timely action on package energy excursions. On the other hand, programming too-small of a time window may not give the PCU enough time to sample energy information and enforce the limit. The minimum value of the 'time window' can be obtained by reading bits [21:15] of the PWR_LIMIT_MISC_INFO CSR using the PECI RdPCIConfigLocal() command.

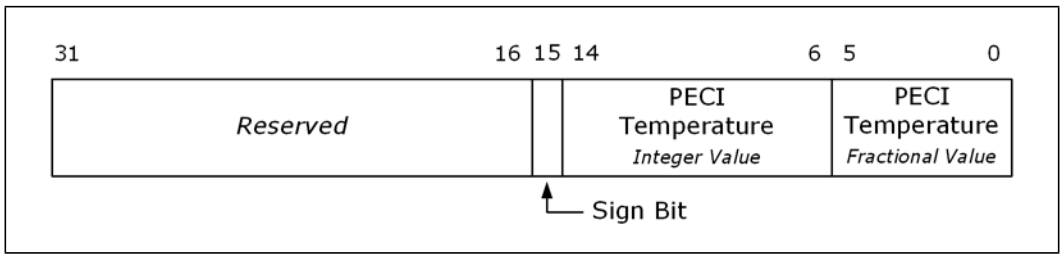
17.9.4 Accumulated Run Time Read (Index = 31)

This read returns the total time for which the processor has been executing with a resolution of 1 ms per count. This is tracked by a 32-bit counter that rolls over on reaching the maximum value. This counter activates and starts counting for the first time when the SoC de-asserts the active-low CPU_RESET_B output signal pin.

17.9.5 Package Temperature Read (Index = 2)

This read returns the maximum processor die temperature in 16-bit PECI format. The upper 16 bits of the response data are reserved. See Figure 17-19. The PECI temperature data returned by this read is the "instantaneous" value and not the "average" value as is returned by the PECI GetTemp() described in Section 17.6.4, "PECI Proxy Command Trigger" on page 389.

Figure 17-19.Package Temperature Read Data



17.9.6 Per Core DTS Temperature Read (Index = 9)

This feature enables the PECI host to read the maximum value of the Digital Thermal Sensor (DTS) temperature for any specific core within the processor. Alternatively, this service can be used to read the internal SoC System Agent (SSA) temperature. The temperature is returned in the same data format as described in Section 17.9.5, "Package Temperature Read (Index = 2)" on page 420. Data is returned in relative PECI temperature format.

Reads to a parameter value outside the supported range return an error as indicated by a completion code of 0x90. The supported range of parameter values can vary depending on the number of processor cores within the SoC. The temperature data returned through this feature is the "instantaneous" value and not the "average" value. It is updated once every 1 ms.

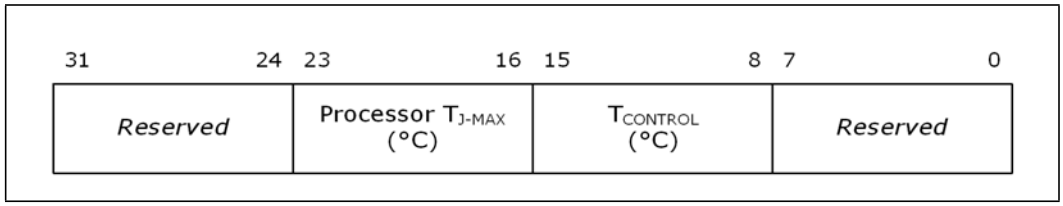
17.9.7 Temperature Target Read (Index = 16)

The Temperature Target Read allows the PECI host to access the maximum Processor Junction Temperature (T_{J-MAX}) in degrees Celsius. This is also the default temperature value at which the processor thermal control circuit activates. The T_{J-MAX} value may vary from processor part to part to reflect manufacturing process variations.

The Temperature Target read also returns the processor $T_{CONTROL}$ value. The $T_{CONTROL}$ is returned in standard PECI temperature format and represents the threshold temperature used by the thermal management system for fan speed control.

See Figure 17-22 on page 422.

Figure 17-20.Temperature Target Read

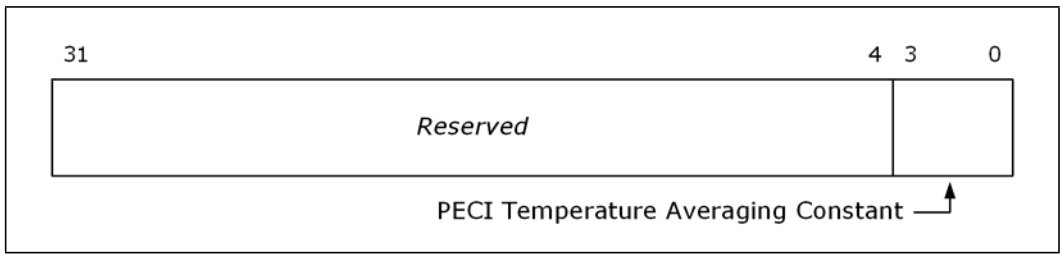


17.9.8 Thermal Averaging Constant Write/Read (Index = 21)

This feature allows the PECI host to control the window over which the estimated processor PECI temperature is filtered. The host may configure this window as a power of two. As an example, programming a value of 5 results in a filtering window of 25 or 32 samples. The maximum programmable value is 8 or 256 samples. Programming a value of zero disables the PECI temperature-averaging feature. The default value of the thermal averaging constant is 4 which translates to an averaging window size of 24 or 16 samples.

See Figure 17-21. Additional details on the PECI temperature filtering function can be found in Section 17.10, "DTS Temperature Data" on page 427.

Figure 17-21.Thermal Averaging Constant Read/Write



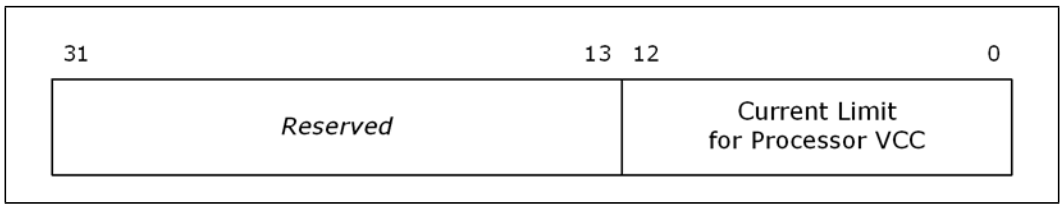
17.9.9 Thermally Constrained Time Read (Index = 32)

This feature allows the PECI host to access the total time for which the processor has been operating in a lowered power state due to Thermal Control Circuit (TCC) activation. The returned data includes the time required to ramp back up to the original P-State target after TCC activation expires. This timer does not include TCC activation as a result of an external assertion of SoC PROCHOT_B signal pin. This is tracked by a 32-bit counter with a resolution of 1 ms per count that rolls over or wraps around. Concerning the processor PECI clients, the only logic that can be thermally constrained are those which are supplied by VCC.

17.9.10 Current Limit Read (Index = 17)

This read returns the current limit for the processor VCC power plane in 1/8-ampere increments. Actual current limit data is contained only in the lower 13 bits of the response data. The default return value of 0x438 corresponds to a current limit value of 135 amperes. See Figure 17-22 on page 422.

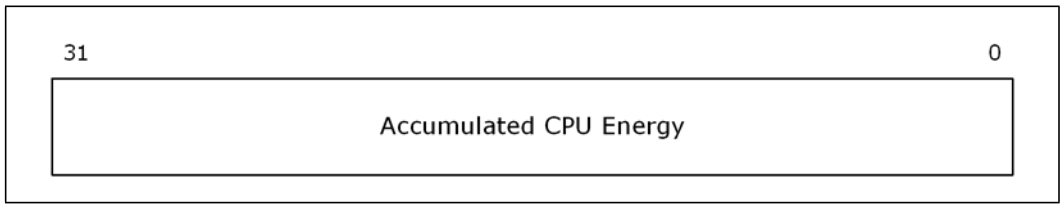
Figure 17-22.Current Limit Read Data



17.9.11 Accumulated Energy Status Read (Index = 3)

This service can return the value of the total energy consumed by the entire processor package or just the logic supplied by the VCC power plane as specified through the parameter field in Table 17-26 on page 412. This information is tracked by a 32-bit counter that wraps around and continues counting on reaching its limit. See Figure 17-23 on page 423. Energy units for this read are determined as per the Package Power SKU Unit settings described in Section 17.9.2, "Package Power SKU Unit Read (Index = 30)" on page 418.

Figure 17-23.Accumulated Energy Read Data



While Intel requires reading the accumulated energy data at least once every 16 seconds to ensure functional correctness, a more realistic polling rate recommendation is once every 250 ms for better accuracy. In general, as the power capability decreases, so will the minimum polling rate requirement. It is recommended that you tune the polling rate to reduce the potential impact on other Power-Management features.

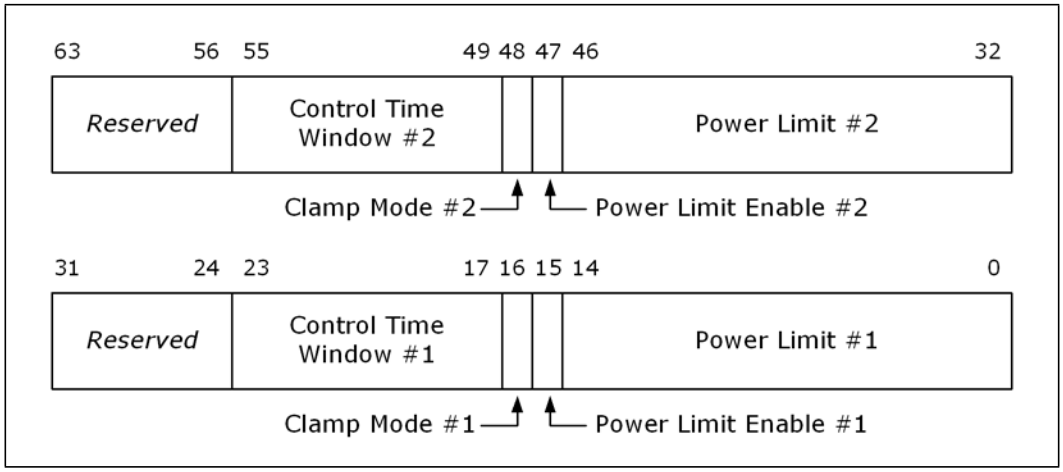
To calculate the power, the following formula can be used:

$$Power(watt) = \frac{EnergyStatus(T_N) - EnergyStatus(T_{N-1})}{(T_N - T_{N-1}) * 2\text{^}energy\_unit}$$

17.9.12 Package Power Limits For Multiple Turbo Modes (Index = 26 and 27)

This feature allows the PECI host to program two power limit values to support multiple turbo modes. The operating systems and drivers can balance the power budget using these two limits. Two separate PECI requests are available to program the lower and upper 32 bits of the power limit data shown in Figure 17-24 on page 424.

Figure 17-24.Package Turbo Power Limit Data



The units for the Power Limit and Control Time Window are determined as per the Package Power SKU Unit settings described in Section 17.9.2, "Package Power SKU Unit Read (Index = 30)" on page 418, while the valid range for power limit values are determined by the Package Power SKU settings described in Section 17.9.2, "Package Power SKU Unit Read (Index = 30)" on page 418 and Section 17.9.3, "Package Power SKU Read (Index = 28 and 29)" on page 419.

Setting the Clamp Mode bits is required to allow the cores to go into power states below what the operating system originally requested. The Power Limit Enable bits should be set to enable the power limiting function. Power Limit values, enable and clamp mode bits can all be set in the same command cycle.

Intel recommends exclusive use of just one entity or interface, PECI for instance, to manage all processor package power limiting and budgeting needs. If PECI is being used to manage package power limiting activities, BIOS should lock out all subsequent in-band package power limiting accesses by setting bit 31 of the MSR_PKG_POWER_LIMIT (MSR 610h) or PKG_TURBO_POWER_LIMIT (SoC sideband Port 04h, offset 7 and 8) register to 1.

Power Limit #1 is intended to limit processor power consumption to any reasonable value below TDP and defaults to TDP. Power Limit #1 values may be impacted by the processor heat sinks and system air flow. Processor Power Limit #2 can be used as appropriate to limit the current drawn by the processor to prevent any external power supply unit issues.

Power Limit #2 should always be programmed to a value (typically 20%) higher than Power Limit #1 and has no default value associated with it.

Though this feature is disabled by default and external programming is required to enable, initialize and control Package Power Limit values and time windows, the processor package will still turbo to TDP if Power Limit #1 is not enabled or initialized.

Control Time Window #1 values may be programmed to be within a range of 100 mS-10 seconds. Control Time Window #2 values should be in the range 10 ms to 100 ms.

The following formula is used to calculate the Power Limit for Power Limit #1 and Power Limit #2:

Power_Limit = Power (in mW) / (2^ POWER_UNIT)

For example, if the Power Limit is 15 Watts, then:

Power_Limit = (15 * 1000) / 2^3

= 1875

The following formula can be used to calculate Control Time Windows (tau), given the following

Time Window = (float) ((1 + (X/4)) * (2 ^ Y))

where X = Z[6:5] and Y = Z[4:0]

The following formula is used to calculate the Control Time Window:

$$Y = \log_2(t_x * 2^{TIMEUNIT})$$

$$Z = 10\left(\left(\frac{t_x * 2^{TIMEUNIT}}{2^Y}\right) - 1\right)$$

$$tau = ((Z \,\&\, 0x3) \ll 4) \mid (Y \,\&\, 0x3F)$$

where t_x is the desired time.

For example, assume a TIMEUNIT of 10 and a desired time-window of 10 seconds:

$$Y = \text{int}(\log_2(10 * 2^{10})) = int(13.3219) = 13 = 0xD$$

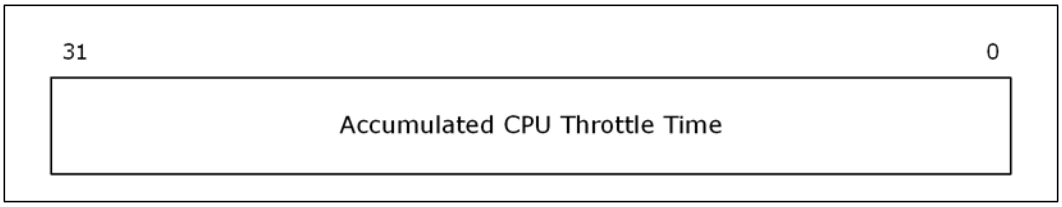
$$Z = int\left(10\left(\left(\frac{10 * 2^{10}}{2^{13}}\right) - 1\right)\right) = int(2.5) = 2 = 0x2$$

$$tau = ((Z \,\&\, 0x3) \ll 4) \mid (Y \,\&\, 0x3F) = 0x2D$$

17.9.13 Package Power Limit Performance Status Read (Index = 8)

This service allows the PECI host to assess the performance impact of the currently active power limiting modes. The read return data contains the total amount of time for which the entire processor package has been operating in a power state that is lower than what the operating system originally requested. This information is tracked by a 32-bit counter that wraps around. See Figure 17-25 on page 426. The unit for time is determined as per the Package Power SKU Unit settings described in Section 17.9.2, "Package Power SKU Unit Read (Index = 30)" on page 418.

Figure 17-25. Package Power Limit Performance Data



17.9.14 Wake-on-PECI Mode Bit Write/Read (Index = 5)

The wake on PECI bit allows for waking up the SoC if in a Package C-State. If it is not set, PECI will not respond to queries when in Package C6. This has no effect in the C2000 Product Family.

17.9.15 SoC Power Budget (Index = 40)

This service proves the overall Thermal Design Power (TDP) for the SoC. It also provides the package consumed power.

Table 17-28. SoC Power Budget Data Format

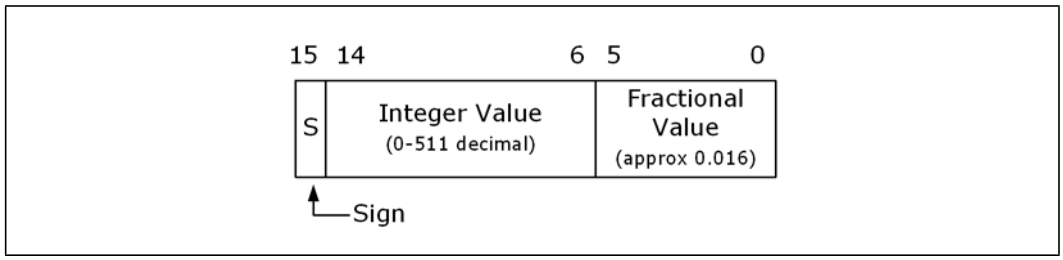
Bits	Access Type	Default Value	Description
31	RO	1'h0	**RESERVED_3**: Reserved.
30:16	RO	15'h0	**SOC_POWER_STS:** Instantaneous SoC power consumption reported by SoC. SoC may use this field to allocate energy credits to SoC and restrict CPU Turbo. If this field is greater than SOC_TDP, package energy credit will be depleted and CPU will not be allowed to Turbo. Power is represented in units specified in PKG_POWER_SKU_UNIT [POWER_UNIT].
15	RO	1'h0	**RESERVED_1**: Reserved.
14:0	RO	15'h0	**SOC_TDP:** TDP budget for SoC and other SoC components. Initialized by the SoC to SOC_TDP_FUSE. Used by Power Budget Manager calculations to determine package energy credit. Value from SOC_TDP_FUSE is scaled according to PKG_POWER_SKU_UNIT [POWER_UNIT].

17.10 DTS Temperature Data

17.10.1 PECI Device Temp Data

When accessed using the PECI GetTemp() command, the temperature is formatted in a 16-bit, 2s complement value representing a number of 1/64 °C. See Figure 17-26. This format allows temperatures in a range of +/-512 °C to be reported to approximately a 0.016 °C resolution.

Figure 17-26.PECI Device Temp [15:0] Format - Temperature Sensor Data



17.10.2 Interpretation

The resolution of the processor Digital Thermal Sensor (DTS) is approximately 1 °C, which can be confirmed by a performing a Read Model-Specific Register (RDMSR) Intel® architecture instruction to the IA32_THERM_STATUS MSR (19Ch)where it is architecturally defined. Note that the MSR read will return only bits [13:6] of the PECI temperature sensor data defined in Figure 17-26 on page 427. PECI temperatures are sent through a configurable low-pass filter prior to delivery in the GetTemp() response data. The output of this filter produces temperatures at the full 1/64 °C resolution even though the DTS itself is not this accurate.

Temperature readings from the processor are always negative in a 2s complement format, and imply an offset from the processor $T_{J\text{-}MAX}$ (PECI = 0). For example, if the processor $T_{J\text{-}MAX}$ is 100 °C, a PECI thermal reading of -10 implies that the processor is running at approximately 10 °C below $T_{J\text{-}MAX}$ which would be 90 °C. PECI temperature readings are not reliable at temperatures above $T_{J\text{-}MAX}$ since the processor is outside its operating range and hence, PECI temperature readings are never positive.

The changes in PECI data counts are approximately linear in relation to changes in temperature in degrees Celsius. A change of 1 in the PECI count represents roughly a temperature change of 1 °C. This linearity is approximate and cannot be guaranteed over the entire range of PECI temperatures, especially as the offset from the maximum PECI temperature (zero) increases.

17.10.3 Temperature Filtering

The processor Digital Thermal Sensor (DTS) provides an improved capability to monitor device hot spots, which inherently leads to more varying temperature readings over short time intervals. Coupled with the fact that typical fan speed controllers may only read temperatures at 4Hz, it is necessary for the thermal readings to reflect thermal trends and not instantaneous readings. Therefore, PECI supports a configurable low-pass temperature filtering function that is expressed by the equation:

$$T_N = (1-\alpha) * T_{N-1} + \alpha * T_{SAMPLE}$$

where T_N and T_{N-1} are the current and previous averaged PECI temperature values respectively, T_{SAMPLE} is the current PECI temperature sample value and the variable 'α' = 1/2*X*, where *X* is the Thermal Averaging Constant that is programmable as described in Section 17.9.8, "Thermal Averaging Constant Write/Read (Index = 21)" on page 421.

17.10.4 Reserved Values

Several values well out of the operational range are reserved to signal temperature sensor errors. These are shown in Table 17-29.

Table 17-29. Error Codes

Error Codes	Description
0x8000	General sensor error.
0x8001	*Reserved*
0x8002	Sensor is operational, but has detected a temperature below its operational range.
0x8003	Sensor is operational, but has detected a temperature above its operational range.
0x8004 – 0x81FF	*Reserved*

§ §

18 SMBus 2.0 Unit 0 - PCU

The SoC provides multiple System Management Bus (SMBus) 2.0 controllers. The SMBus controller described in this chapter is located in the Platform Control Unit (PCU) of the SoC. In SoC diagrams, it is labeled SMBus0.

The host controller provides a mechanism for the processor to initiate communications with SMBus peripherals (slaves). The SoC is also capable of operating in a mode that communicates with I^2C-compatible devices.

The SoC performs SMBus messages with Packet Error Checking (PEC) enabled or disabled. The actual PEC calculation and checking are performed in either the hardware or the software.

The SMBus Address Resolution Protocol (ARP) is supported by using the existing host controller commands through the software, except for the Host Notify command (which is actually a received message).

Figure 18-1. SMBus PCU Covered in This Chapter

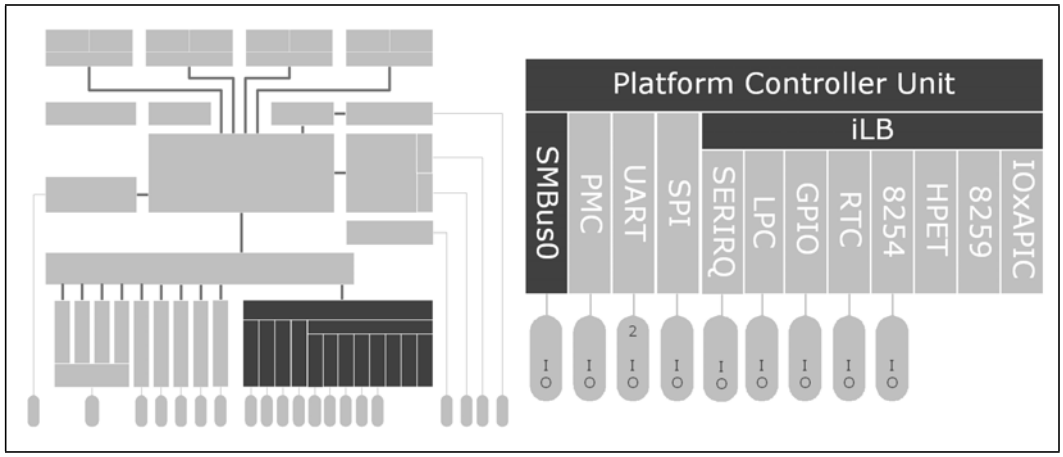


Table 18-1. References

Reference	Revision	Date	Document Title
SMBus	2.0	August 3, 2000	*System Management Bus (SMBus) Specification,* Version 2.0

18.1 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

Table 18-2. Signal Names

Signal Name	Direction Type	Description
SMB_CLK0	I/OD	**SMBus Clock (SMBCLK):** *This signal is muxed with GPIOS_9 and is used by other functions.*
SMB_DATA0	I/OD	**SMBus Data (SMBDAT):** *This signal is muxed with GPIOS_8 and is used by other functions.*
SMBALRT_N0	I/OD	**SMBus Alert (SMBALERT#):** This signal wakes the system or generates a System Management Interrupt (SMI). *This signal is muxed with GPIOS_10 and is used by other functions.*

The optional SMBus 2.0 signal, SMBSUS#, is not supported.

18.2 General Architecture

At its network layer, the *System Management Bus Specification* refers to three types of devices:

- Slave Device - A device that is receiving or responding to a command.
- Master - A device that issues commands, generates the clocks, and terminates the transfer.
- Host - A specialized master that provides the main interface to the system CPU. A host must be a master-slave and must support the SMBus host notify protocol. At most, one host exists in a system.

The SMBus controller described in this chapter is a system host. It is an example of an SMBus device that acts as a host most of the time but that includes some slave-device behavior.

The programming model of the host controller is combined into two portions: a PCI configuration portion and a system I/O mapped portion. All static configurations, such as the I/O base address, is done using the PCI configuration space. Real-time programming of the host interface is done in the system I/O space.

The SMBus interface is disabled by setting FUNC_DIS_2.SMB_DIS to 1b.

18.3 System Host Controller

The SMBus host controller sends commands to other SMBus slave devices. The software sets up the host controller with an address, command, and for writes, data and optional Packet Error Checking (PEC); and then tells the controller to start. When the controller has finished transmitting data on writes or receiving data on reads, it generates an INTB or a System Management Interrupt (SMI) depending on how the software has configured the controller.

The host controller supports eight command protocols of the SMBus interface. See the *System Management Bus (SMBus) Specification,* Version 2.0. They are:

- Quick Command
- Send Byte
- Receive Byte
- Write Byte/Word
- Read Byte/Word
- Process Call
- Block Read/Write
- Block Write-Block Read Process Call

Additionally, it supports one command protocol for I^2C devices:

- I^2C Read

The SMBus host controller requires that the various data and command fields be setup for the type of command to be sent. When the software sets the START bit, the SMBus host controller performs the requested transaction and interrupts the processor (or generates an SMI) when the transaction is completed. Once a START command has been issued, the values of the active registers [Host Control (SMB_Mem_HCTL), Host Command (SMB_Mem_HCMD), Transmit Slave Address (SMB_Mem_TSA), Data 0 (SMB_Mem_HD0), Data 1 (SMB_Mem_HD1)] are not changed or read until the interrupt status message (SMB_Mem_HSTS.INTR) has been set (indicating the completion of the command). Any register values needed for computation purposes are saved before issuing of a new command since the SMBus host controller updates all registers while completing the new command.

18.3.1 Command Protocols

In all of the following commands, the Host Status (SMB_Mem_HSTS) register determines the progress of the command. While the command is in operation, the SMB_Mem_HSTS.HBSY bit is set. If the command completes successfully, the SMB_Mem_HSTS.INTR bit is set. If the device does not respond with an acknowledge, and the transaction times out, the SMB_Mem_HSTS.DEVERR bit is set. If the software sets the SMB_Mem_HCTL.KILL bit while the command is running, the transaction stops and the SMB_Mem_HSTS.FAILED bit is set.

18.3.1.1 Quick Command

When programmed for a Quick Command, the Transmit Slave Address (SMB_Mem_TSA) register is sent. The PEC byte is never appended to the quick protocol. The software forces the SMB_Config_HCTL.PECEN bit to 0b when performing the Quick Command. The software must force the SMB_Config_HCFG.I2C_EN bit to 0b when running this command. See Section 5.5.1 of the System Management Bus (SMBus) Specification, Version 2.0 for the format of the protocol.

18.3.1.2 Send Byte/Receive Byte Command

For the Send Byte command, the Transmit Slave Address (SMB_Mem_TSA) and Host Command (SMB_Mem_HCMD) registers are sent. For the Receive Byte command, the Transmit Slave Address (SMB_Mem_TSA) register is sent. The data received is stored in the Data 0 (SMB_Mem_HD0) register. The software must force the SMB_Config_HCFG.I2C_EN bit to 0b when running this command.

The Receive Byte is similar to a Send Byte, the only difference is the direction of data transfer. See Sections 5.5.2 and 5.5.3 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

18.3.1.3 Write Byte/Word Command

The first byte of a Write Byte/Word access is the command code. The next 1 or 2 bytes are the data to be written. When programmed for a Write Byte/Word command, the Transmit Slave Address (SMB_Mem_TSA), Host Command (SMB_Mem_HCMD), and Data 0 (SMB_Mem_HD0) registers are sent. In addition, the Data 1 (SMB_Mem_HD1) register is sent on a Write Word command. The software must force the SMB_Config_HCFG.I2C_EN bit to 0 when running this command. See Section 5.5.4 of the System Management Bus (SMBus) Specification, Version 2.0 for the format of the protocol.

18.3.1.4 Read Byte/Word Command

Reading data is slightly more complicated than writing data. First the SoC must write a command to the slave device. Then it must follow that command with a repeated start condition to denote a read from that device address. The slave then returns 1 or 2 bytes of data. The software must force the SMB_Config_HCFG.I2C_EN bit to 0b when running this command.

When programmed for the Read Byte/Word command, the Transmit Slave Address (SMB_Mem_TSA) and Host Command (SMB_Mem_HCMD) registers are sent. Data is received into the Data 0 (SMB_Mem_HD0) on the read byte, and the Data 0 (SMB_Mem_HD0) and Data 1 (SMB_Mem_HD1) registers on the read word. See Section 5.5.5 of the System Management Bus (SMBus) Specification, Version 2.0 for the format of the protocol.

18.3.1.5 Process Call Command

The Process Call is so named because a command sends data and waits for the slave to return a value dependent on that data. The protocol is a Write Word followed by a Read Word, but without a second command or stop condition.

When programmed for the Process Call command, the SoC transmits the Transmit Slave Address (SMB_Mem_TSA), Host Command (SMB_Mem_HCMD), Data 0 (SMB_Mem_HD0) and Data 1 (SMB_Mem_HD1) registers. Data received from the device is stored in the Data 0 (SMB_Mem_HD0) and Data 1 (SMB_Mem_HD1) registers. The Process Call command with SMB_Config_HCFG.I2C_EN set and the SMB_Config_HCTL.PECEN bit set produces undefined results. The software must force either SMB_Config_HCFG.I2C_EN or SMB_Config_HCTL.PECEN and SMB_Mem_AUXC.AAC to 0b when running this command. See Section 5.5.6 of the System Management Bus (SMBus) Specification, Version 2.0 for the format of the protocol.

Note: For the Process Call command, the value written into SMB_Mem_TSA.RW needs to be 0b.

Note: If the SMB_Config_HCFG.I2C_EN bit is set, the protocol sequence changes slightly: the command code (bits [18:11] in the bit sequence) is not sent, and as a result, the slave does not acknowledge (bit 19 in the sequence).

18.3.1.6 Block Read/Write Command

The SoC contains a 32-byte buffer for read and write data which are enabled by setting SMB_Mem_AUXC.E32B, as opposed to a single byte of buffering. This 32-byte buffer is filled with write data before transmission and filled with read data on reception. In the SoC, the interrupt is generated only after a transmission or reception of 32 bytes, or when the entire byte count has been transmitted/received.

Note: When operating in I2C mode (SMB_Config_HCFG.I2C_EN bit is set), the SoC never uses the 32-byte buffer for any block commands.

The byte count field is transmitted but ignored by the SoC as the software ends the transfer after all bytes it cares about have been sent or received.

For a Block Write, the software must either force the SMB_Config_HCFG.I2C_EN bit or both the SMB_Config_HCTL.PECEN and SMB_Mem_AUXC.AAC bits to 0b when running this command.

The Block Write begins with a slave address and a write condition. After the command code, the SoC issues a byte count describing how many more bytes follow in the message. If a slave has 20 bytes to send, the first byte is the number 20 (14h), followed by 20 bytes of data. The byte count is not 0. A Block Read or Write is allowed to transfer a maximum of 32-data bytes.

When programmed for a Block Write command, the Transmit Slave Address (SMB_Mem_TSA), Host Command (SMB_Mem_HCMD) and Data 0 (SMB_Mem_HD0) registers are sent. Data is then sent from the Host Block Data (SMB_Mem_HBD) register; the total data sent being the value stored in the Data 0 (SMB_Mem_HD0) register. On Block Read commands, the first byte received is stored in the Data 0 (SMB_Mem_HD0) register, and the remaining bytes are stored in the Host Block Data (SMB_Mem_HBD) register. See Section 5.5.7 of the System Management Bus (SMBus) Specification, Version 2.0 for the format of the protocol.

Note: For a Block Write, if the SMB_Config_HCFG.I2C_EN bit is set, the format of the command changes slightly. The SoC still sends the number of bytes (on writes) or receive the number of bytes (on reads) indicated in the Data 0 (SMB_Mem_HD0) register. However, it does not send the contents of the Data 0 (SMB_Mem_HD0) register as part of the message. Also, the block write protocol sequence changes slightly: the byte count (bits [27:20] in the bit sequence) is not sent, and as a result, the slave does not acknowledge (bit 28 in the sequence).

18.3.1.7 Block Write-Block Read Process Call Command

The Block Write-Block Read Process Call command is a two-part message. The call begins with a slave address and a write condition. After the command code, the host issues a write byte count (M) that describes how many more bytes are written in the first part of the message. If a master has 6 bytes to send, the byte count field has the value 6 (0000 0110b), followed by the 6 bytes of data. The write byte count (M) cannot be 0.

The second part of the message is a block of read data beginning with a repeated start condition followed by the slave address and a read bit. The next byte is the read byte count (N), which differs from the write byte count (M). The read byte count (N) cannot be 0.

The combined data payload must not exceed 32 bytes. The byte-length restrictions of this process call are summarized as follows:

- M ≥ 1 byte
- N ≥ 1 byte
- M + N ≤ 32 bytes

The read byte count does not include the Packet Error Checking (PEC) byte. The PEC is computed on the total message beginning with the first slave address and using the normal PEC computational rules. Intel recommends that a PEC byte be used with the Block Write-Block Read Process Call command. The software must do a read to the Host Command (SMB_Mem_HCMD) register to reset the 32-byte buffer pointer before reading the Host Block Data (SMB_Mem_HBD) register.

Note: No STOP condition is before the repeated START condition, and that a NACK signifies the end of the read transfer.

Note: The SMB_Mem_AUXC.E32B bit in the Auxiliary Control register must be set when using this protocol.

See Section 5.5.8 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

18.3.1.8 I²C Read Command

This command allows the SoC to perform block reads to certain I²C devices, such as serial EEPROMs. The SMBus Block Read supports the 7-bit addressing mode only. However, this does not allow access to devices using the I²C combined format that has data bytes after the address. Typically these data bytes correspond to an offset (address) within the serial memory chips.

Note: This command is supported independent of the setting of the SMB_Config_HCFG.I2C_EN bit. The I2C Read command with the SMB_Config_HCTL.PECEN bit set produces undefined results. The software must force both the SMB_Config_HCTL.PECEN and SMB_Mem_AUXC.AAC bit to 0b when running this command.

For an I²C Read command, the value written into SMB_Mem_TSA.RW needs to be 1b. The format that is used for the command is shown in Table 18-3.

Table 18-3. I²C Block Read

Bit	Description
1	Start
8:2	Slave address – 7 bits
9	Write
10	Acknowledge from slave
18:11	Send Data 1 (SMB_Mem_HD1) register
19	Acknowledge from slave
20	Repeated start
27:21	Slave address – 7 bits
28	Read
29	Acknowledge from slave
37:30	Data byte 1 from slave – 8 bits
38	Acknowledge
46:39	Data byte 2 from slave – 8 bits
47	Acknowledge
–	Data bytes from slave/acknowledge
–	Data byte N from slave – 8 bits
–	NOT acknowledge
–	Stop

The SoC continues reading data from the peripheral until the NAK is received.

18.3.2 Bus Arbitration

Several masters attempt to get on the bus at the same time by driving the SMBDAT (the SMB_DATA0 signal in this chapter) line low to signal a start condition. The SoC continuously monitors the SMBDAT line. When the SoC is attempting to drive the bus to a 1 by letting go of the SMBDAT line, and it samples SMBDAT low, then some other master is driving the bus and the SoC stops transferring data.

If the SoC sees that it has lost arbitration, the condition is called a collision. The SoC sets SMB_Mem_HSTS.BERR, and if enabled, generates an interrupt or SMI. The processor restarts the transaction.

The SoC, as a SMBus master, drives the clock. When the SoC is sending an address or a command or data bytes on writes, it drives the data relative to the clock it is also driving. It does not start toggling the clock until the start or stop condition meets the proper setup and hold time. The SoC also ensures minimum time between the SMBus transactions as a master.

18.3.3 Bus Timing

18.3.3.1 Clock Stretching

Some devices are not able to handle their clock toggling at the rate that the SoC as an SMBus master is currently. They can stretch the low time of the clock. When the SoC attempts to release the clock (allowing the clock to go high), the clock remains low for an extended period of time.

The SoC monitors the SMBus clock line after it releases the bus to determine whether to enable the counter for the high time of the clock. While the bus is still low, the high time counter must not be enabled. Similarly, the low period of the clock is stretched by an SMBus master if it is not ready to send or receive data.

18.3.3.2 Bus Time Out (the SoC as SMBus Master)

If an error is in the transaction, such that an SMBus device does not signal an acknowledge, or holds the clock lower than the allowed time-out time, the transaction times out. The SoC discards the cycle and sets the SMB_Mem_HSTS.DEVERR bit. The time out minimum is 25 ms (800 RTC clocks). The time-out counter inside the SoC starts after the last bit of data is transferred by the SoC, and it is waiting for a response.

The 25 ms time-out counter does not count under the following conditions:

1. The SMB_Mem_HSTS.BYTE_DONE_STS bit is set.
2. The TCO_STS.SECOND_TO_STS bit is not set (this indicates that the system has not locked up).

18.3.4 Interrupts and SMI

The SMBus controller uses INTB as its virtual interrupt wire. However, the system is alternatively set up to generate a System Management Interrupt (SMI) instead of an interrupt, by setting the SMB_Config_HCFG.SMI_EN bit.

Table 18-4 and Table 18-5 specify how the various enable bits in the SMBus function control the generation of the interrupt, host SMI, and wake internal signals. The rows in the tables are additive, which means that if more than one row is true for a particular scenario then the results for all of the activated rows occur.

Table 18-4. Enable for SMBALRT_N

Event	SMB_Mem_HCTL.INTREN	SMB_Config_HCFG.SMI_EN	SMB_Mem_SCMD.SMBALTDIS	Result
SMBALRT_N (always reported in SMB_Mem_HSTS.SMBALERT)	X	1	0	Slave SMI generated (SMBUS_SMI_STS)
	1	0	0	Interrupt generated

Table 18-5. Enables for SMBus Host Events

Event	SMB_Mem_HCTL.INTREN	SMB_Config_HCFG.SMI_EN	Event
Any combination of SMB_Mem_HSTS.FAILED, SMB_Mem_HSTS.BERR, SMB_Mem_HSTS.DEVERR, SMB_Mem_HSTS.INTR asserted	0	X	None
	1	0	Interrupt generated
	1	1	Host SMI generated

Table 18-6. Enables for the Host Notify Command

SMB_Mem_SCMD.HNINTREN	SMB_Config_HCFG.SMI_EN	SMB_Mem_SCMD.HNWAKEEN	Result
0	X	0	None
1	0	X	Interrupt generated
1	1	X	Slave SMI generated (SMBUS_SMI_STS)

18.3.5 SMBALRT_N

The SMBALRT_N signal is multiplexed with GPIOS_10. If the SMBALRT_N is not blocked by SMB_Mem_SCMD_io, when it is asserted, the SoC generates an interrupt or an SMI.

Note: Using this signal as a wake event from S5 is not supported.

18.3.6 SMBus CRC Generation and Checking

If the SMB_Mem_AUXC.AAC is set, the SoC automatically calculates and drives the Cyclic Redundancy Check (CRC) at the end of the transmitted packet for write cycles and checks the CRC for read cycles. It does not transmit the contents of the Packet Error Check Data (SMB_Mem_PEC) PEC register for a CRC. The SMB_Mem_HCTL.PECEN bit must not be set if this bit is set, or unspecified behavior results.

If the read cycle results in a CRC error, the SMB_Mem_HSTS.DEVERR bit and the SMB_Mem_AUXS.CRCE bit are set.

18.4 SMBus Slave Interface

The SoC does not implement a complete SMBus slave interface. Only the Host Notify command is implemented to maintain specification compatibility.

18.4.1 Host Notify Command Format

The SoC tracks and responds to the standard Host Notify command as specified in the System Management Bus (SMBus) Specification, Version 2.0. The host address for this command is fixed to 0001000b. If the SoC already has data for a previously-received Host Notify command which has not been serviced yet by the host software (as indicated by the SMB_Mem_SSTS.HNST bit), then it does NACK following the host address byte of the protocol. This allows the host to communicate non-acceptance to the master and retain the host notify address and data values for the previous cycle until the host software completely services the interrupt.

Note: The host software must always clear the SMB_Mem_SSTS.HNST bit after completing any necessary reads of the address and data registers. Table 18-7 shows the Host Notify command format.

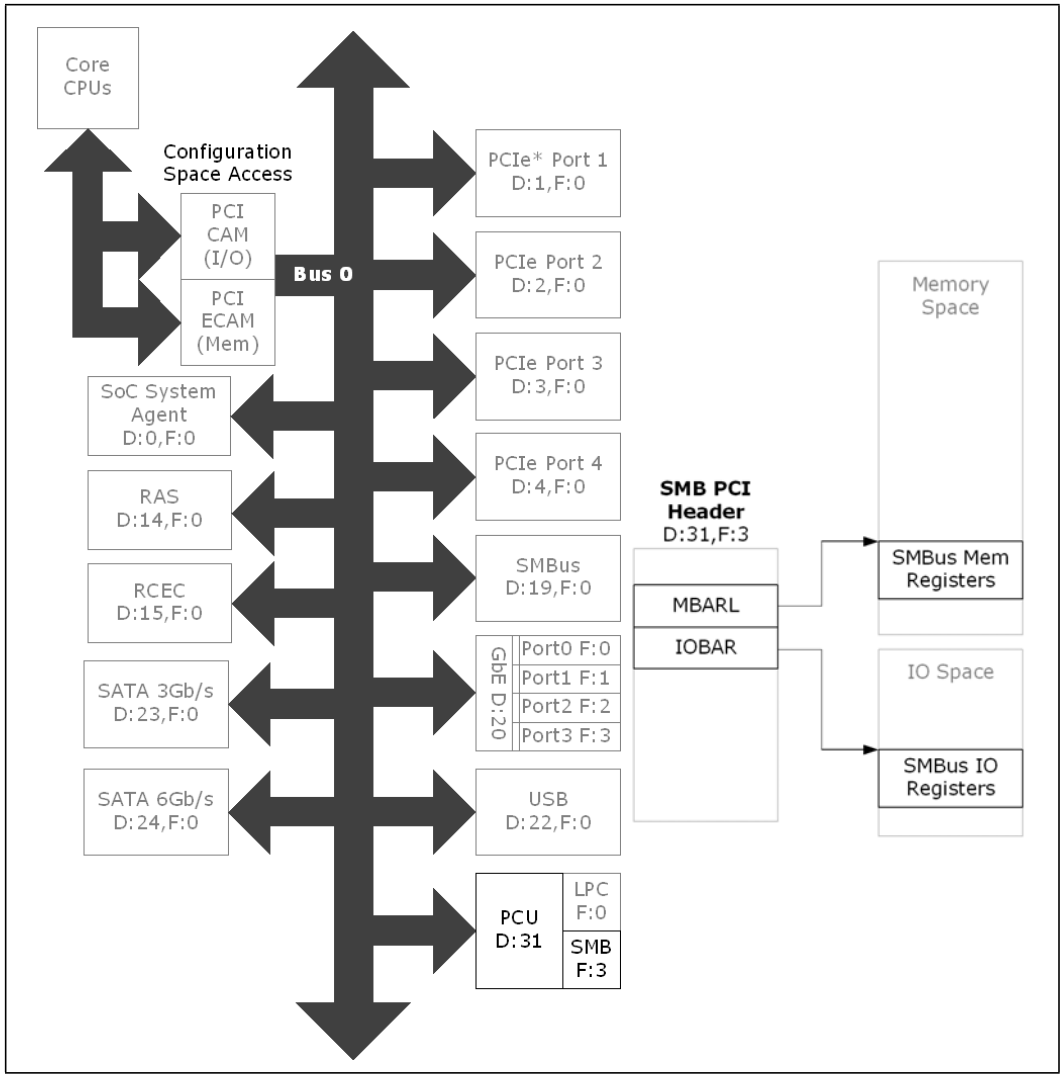
Table 18-7. Host Notify Command Format

Bit	Description	Driven By	Comment
1	Start	External Master	
8:2	SMB Host Address – 7 bits	External Master	Always 0001_000
9	Write	External Master	Always 0
10	ACK (or NACK)	SoC	SoC NACKs if SMB_Mem_SSTS.HNST is 1
17:11	Device Address – 7 bits	External Master	Indicates the address of the master; loaded into the Notify Device Address register (SMB_Mem_NDA)
18	Unused – Always 0	External Master	7-bit only address; this bit is inserted to complete the byte
19	ACK	SoC	
27:20	Data Byte Low – 8 bits	External Master	Loaded into the Notify Data Low Byte register (SMB_Mem_NDLB)
28	ACK	SoC	
36:29	Data Byte High – 8 bits	External Master	Loaded into the Notify Data High Byte register (SMB_Mem_NDLB)
37	ACK	SoC	
38	Stop	External Master	

18.5 Register Map

Figure 18-2 shows the SoC PCU-SMBus 2.0 Host Controller registers from a system viewpoint.

Figure 18-2. PCU-SMBus 2.0 Register Map



18.5.1 Registers in Configuration Space

The list of the PCU-SMBus 2.0 registers in the configuration space is shown in Table 18-8. The registers are in the configuration space starting at bus 0, device 31 (decimal), function 3. The offset addresses are listed.

Table 18-8. PCU-SMBus 2.0 Registers in Configuration Space

Configuration Address Offset	Name	Description
0x00	SMB_Config_VID	D31_F3_Vendor ID
0x02	SMB_Config_DID	D31_F3_Device ID
0x04	SMB_Config_CMD	D31_F3_Command
0x06	SMB_Config_STAT	D31_F3_Device_Status
0x08	SMB_Config_REV	D31_F3_Revision ID
0x09	SMB_Config_PRGIF	D31_F3_Programming Interface
0x0A	SMB_config_SCC	D31_F3_Sub Class Code
0x0B	SMB_Config_BCC	D31_F3_Base Class Code
0x10	SMB_Config_MBARL	D31_F3_SMBus Memory Base Address
0x14	SMB_Config_MBARH	D31_F3_SMBus Memory Base Address
0x20	SMB_Config_IOBAR	D31_F3_SMB I/O Base Address
0x2C	SMB_Config_SVID	D31_F3_SVID
0x2E	SMB_Config_SID	D31_F3_SID
0x3C	SMB_Config_INTLN	D31_F3_Interrupt Line
0x3D	SMB_Config_INTPN	D31_F3_Interrupt Pin
0x40	SMB_Config_HCFG	D31_F3_Host Configuration
0xF0	SMB_Config_ERR	IOSF Error Control
0xF8	SMB_Config_MANID	D31_F3_Manufacturer's ID

18.5.2 Registers in Memory Space

The list of the PCU-SMBus 2.0 registers in the memory space is shown in Table 18-9. This list of MMIO registers starts at the memory address designated by the 32-bit MBARL register listed in Table 18-8 on page 442. The MBARH register is not used and the MMIO must be in the 32-bit addressing space.

Table 18-9. PCU-SMBus 2.0 Registers in Memory Space

Memory Address Offset	Name	Description
0x00	SMB_Mem_HSTS	Host Status Register
0x02	SMB_Mem_HCTL	Host Control Register
0x03	SMB_Mem_HCMD	Host Command Register
0x04	SMB_Mem_TSA	Transmit Slave Address Register
0x05	SMB_Mem_HD0	Data 0 Register
0x06	SMB_Mem_HD1	Data 1 Register
0x07	SMB_Mem_HBD	Host Block Data
0x08	SMB_Mem_PEC	Packet Error Check Data Register
0x09	SMB_Mem_SADDR	Receive Slave Address Register
0x0C	SMB_Mem_AUXS	Auxiliary Status
0x0D	SMB_Mem_AUXC	Auxiliary Control
0x0E	SMB_Mem_SMLC	SMLINK_PIN_CTL Register (not supported)
0x0F	SMB_Mem_SMBC	SMBUS_PIN_CTL Register
0x10	SMB_Mem_SSTS	Slave Status Register
0x11	SMB_Mem_SCMD	Slave Command Register
0x14	SMB_Mem_NDA	Notify Device Address Register
0x16	SMB_Mem_NDLB	Notify Data Low Byte Register
0x17	SMB_Mem_NDHB	Notify Data High Byte Register

18.5.3 Registers in I/O Space

The list of the PCU-SMBus 2.0 registers in the I/O space is shown in Table 18-10. This list of I/O registers starts at the I/O address designated by the 32-bit IOBAR register listed in Table 18-8 on page 442.

Table 18-10. PCU-SMBus 2.0 Registers in I/O Space

I/O Address Offset	Name	Description
0x00	SMB_Mem_HSTS_io	Host Status Register
0x02	SMB_Mem_HCTL_io	Host Control Register
0x03	SMB_Mem_HCMD_io	Host Command Register
0x04	SMB_Mem_TSA_io	Transmit Slave Address Register
0x05	SMB_Mem_HD0_io	Data 0 Register
0x06	SMB_Mem_HD1_io	Data 1 Register
0x07	SMB_Mem_HBD_io	Host Block Data
0x08	SMB_Mem_PEC_io	Packet Error Check Data Register
0x09	SMB_Mem_SADDR_io	Receive Slave Address Register
0x0C	SMB_Mem_AUXS_io	Auxiliary Status
0x0D	SMB_Mem_AUXC_io	Auxiliary Control
0x0E	SMB_Mem_SMLC_io	SMLINK_PIN_CTL Register
0x0F	SMB_Mem_SMBC_io	SMBUS_PIN_CTL Register
0x10	SMB_Mem_SSTS_io	Slave Status Register
0x11	SMB_Mem_SCMD_io	Slave Command Register
0x14	SMB_Mem_NDA_io	Notify Device Address Register
0x16	SMB_Mem_NDLB_io	Notify Data Low Byte Register
0x17	SMB_Mem_NDHB_io	Notify Data High Byte Register

§ §

19 Power Management Controller (PMC)

The Power Management Controller (PMC) interfaces with the external circuitry of the platform board. Together they provide the power management functions for the SoC and platform components.

Figure 19-1. Power Management Controller Covered in This Chapter

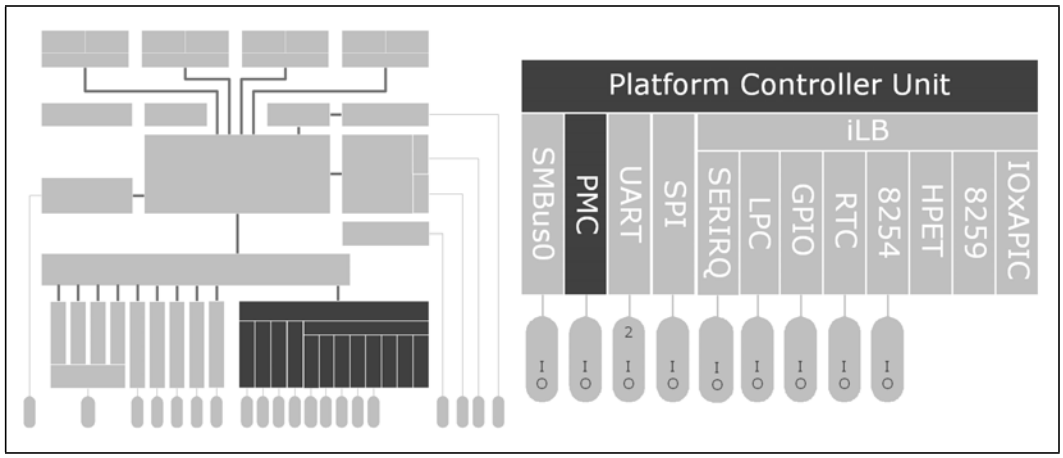


Table 19-1. References

Reference	Revision	Date	Document Title
ACPI Specification	5.0	December 6, 2011	*Advanced Configuration and Power Interface Specification*, Revision 5.0

The *ACPI Specification* is available on the internet at this URL: http://www.acpi.info/.

19.1 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Voltage Level:** Typical operating voltage of the signal
- **Power Well:** SoC power well used for signal circuitry
- **Function:** A brief explanation of the signal functions

Detailed signal descriptions are in Chapter 31, "Signal Names and Descriptions" and multiplexed signal maps are in Chapter 33, "Signal Electrical and Timing Characteristics".

Table 19-2. PMC Signals

Signal Name	Direction	Voltage Level	Power Well	Function
SUSPWRDNACK	Output	3.3V	SUS	Signals the platform board to power down the Suspend (SUS) well during S5. *This signal is muxed and is used by other functions.*
PMU_SLP_DDRVTT_B	Output	3.3V	SUS	Signals the platform board to power down DDR VTT. *This signal is muxed and is used by other functions.*
PMU_SLP_S45_B	Output	3.3V	SUS	Signals the platform board to get into the S5 state and to power down the core well including the DRAM power.
PMU_SLP_S3_B	Output	3.3V	SUS	Signals the platform board that the SoC has entered the S3 state and that the SoC core-well supplies can be powered down. The DRAM power must remain on
PMU_SLP_LAN_B	Output	3.3V	SUS	Signals the platform board to power down the powering to the Gigabit Ethernet PHY circuitry. *This signal is non-functional and is always deasserted: logic high state.*
PMU_PLTRST_B	Output	3.3V	SUS	Used as the platform board reset.
SUS_STAT_B	Output	3.3V	SUS	Indicates to the platform board that a low-power state (S5) is entered soon. The platform needs to work from the SUS well only. *This signal is muxed and is used by other functions.*
PMU_SUSCLK	Output	3.3V	SUS	Suspend clock output. Frequency of 32.768 kHz originating from the Real Time Clock (RTC) clock circuitry. *This signal is muxed and is used by other functions.*
PMU_WAKE_B	Input	3.3V	SUS	Wake signal from the PCI Express* interface. *This signal is muxed and is used by other functions.*
PMU_PWRBTN_B	Input	3.3V	SUS	Power button input. *This signal is muxed and is used by other functions.*
PMU_RESETBUTTON_B	Input	3.3V	Core	Reset button input. *This signal is muxed and is used by other functions.*

19.2 Features

The PMC provides the SoC with these features and functions:

- Power-up sequencing.
- Sleep-state sequencing.
- Global and host partition reset sequencing.
- Keeps the controller S0-state run-time code in integrated RAM.
- Provides SMI and SCI interface and sequencing with the CPU.
- Host TCO watchdog timer.
- Dynamic power management control.
- Lock mechanism for the integrated USB 2.0 ports.

19.3 Architectural Overview

Most of the PMC circuitry and registers are powered by the Suspend (SUS) power well. A small portion of its registers are in the RTC power well. The circuitry for the reset-button input signal is in the core power well.

The SUS power well contains:

- Logic circuitry that is first to become active and powers-up the rest of the SUS well circuitry.
- The PMC microprocessor, its internal code ROM, RAM, and registers.
- Logic circuits that are needed during the sleep states.
- PMC registers accessed by the CPU in the I/O and memory space.
- Legacy CPU watchdog timers.

The RTC power well contains:

- Certain register bits.
- Logic circuitry to generate the power-OK signals.

The core power well contains:

- Reset button input.

The software interfaces with the PMC through a number of registers in the I/O and memory space.

Table 19-3. PMC Register Summary

Addressing Space	Fixed Addressing or BAR	Address or Address Bits	Size of Data Block	Purpose
Memory	PBASE	9-bit BAR	512 bytes	Memory used by the PMC circuitry. The software configuration and status bits for power management and SMI control.
I/O	Fixed 16 bits	0x0092	1 byte	INIT control.
I/O	Fixed 16 bits	0x0CF9	1 byte	Host reset control.
I/O	ABASE	7-bit BAR	96 bytes	ACPI registers for wake and SMI control.
			16 bytes	TCO timer registers.

19.3.1 Reset Behavior

19.3.1.1 Overview

There are numerous sources that can cause the SoC to reset the platform. There are also numerous types of resets that can result. Table 19-4 and the reset type list that follows describe these sources and the SoC reaction. See Chapter 7, "SoC Reset and Power Supply Sequences" for the SoC hardware-signal interface for power, resets, and state transitions.

Table 19-4. SoC Reset Sources

Trigger	Description	Type of Reset (See List below)
Write of 0Eh to CF9h Register	A write of 0Eh to the CF9h register	2
Write of 06h to CF9h Register	A write of 06h to the CF9h register	1
PMU_RESETBUTTON_B and CF9h Bit 3 = 0	The user presses the reset button causing the CPU_RESET_B pin to go active (after the debounce logic).	1
PMU_RESETBUTTON_B and CF9h Bit 3 = 1	The user presses the reset button causing the CPU_RESET_B pin to go active (after the debounce logic).	2
TCO Watchdog Timer	The TCO timer reaches zero two times.	1
Power Failure	The COREPWROK signal goes inactive in S0.	4
S5	The SoC is reset when going into the S5 state.	3
SoC Internal Thermal Trip	The internal thermal sensor signals a catastrophic temperature condition— transition to S5 and reset asserts.	5
PMU_PWRBTN_B (Power Button Override)	A 4-second press causes a transition to S5 (and reset asserts).	5
CPU Shutdown with Policy to Assert PMU_PLTRST_B	A shutdown special cycle from the CPU can cause either INIT or CF9h-style PLTRST.	4; if the CF9h Global Reset Bit = 1b, else 2; if the CF9h Register Bit 3 = 1b, else 1
Write of 06h or 0Eh to CF9h Register	CF9h Global Reset Bit = 1b	4
Host Partition Reset Entry Timeout	The host partition reset entry sequence took longer than the allowed time out value (presumably due to a failure to receive one of the internal or external handshakes).	4
S5 Entry Timeout	An S5 entry sequence took longer than the allowed time out value (presumably due to a failure to receive one of the internal or external handshakes).	5
PMC Watchdog Timer	A firmware hang watchdog time out is detected in the PMC platform.	5

The types of resets are:

1. Host Reset Without Power Cycle (Warm Reset)
   a. The host-only functionality in the SoC gets reset. Any functionality that needs to remain operational during a host reset must not get reset in this case. The SoC is allowed to drop this type of reset request if received while the system is in S5.
2. Host Reset With Power Cycle (Cold Reset)
   a. The host-only functionality in the SoC gets reset. Any functionality that needs to remain operational during a host reset must not get reset in this case. The host system automatically is powered back up and brought out of reset. The SoC must not drop this type of reset request if received while the system is in a software-entered S5 state. If the system is in S5 due to a reset type #5 event, the SoC is allowed to drop this type of reset request.
3. Sx Entry and Host Stays There (Power-Down to S5 Soft Off)
   a. Any functionality that needs to remain operational or retain status during a host-sleep state must not get reset in this case. In this case, the host does not automatically power back up.
4. Global, Power Cycle Reset
   a. The hardware entity is not dependent on the firmware (note that the firmware may initiate the reset). All SoC functionality should get reset, except the:
   — RTC power well backed information.
   — Suspend well status, configuration, and functional logic for controlling and reporting this reset.
   b. The host powers back up after the power cycle period.
5. Straight to S5 and the Host Stays There
   a. All power wells that are controlled by the PMU_SLP_S45_B pins are turned off.
   b. All SoC functionality is reset, except the:
   — RTC power well backed information.
   — Suspend well status, configuration, and functional logic for controlling and reporting this reset.

19.3.2 PMC Memory Area

The PMC contains a 512-byte memory it uses for various power management functions and control. The area is located in 32-bit addressed memory space starting the Base Address Register (BAR) PMC Base Address (PBASE). The BAR is located in the configuration space for the ILB PCI device at offset 44h of bus 0, device 31, function 0. The memory area is not prefetchable. Software access to the 512-byte area is controlled by the Enable (EN) bit of the PBASE pointer.

The PMC registers are shown in Table 19-5.

Table 19-5. PCM Registers in Memory Space

Memory Offset from PBASE	Name	Description
0x00	PRSTS	Power and Reset Status
0x08	PMC_CFG	Power Management Configuration
0x0C	VLV_PM_STS	Power Management Status
0x10	MTPMC	Message to PMC
0x20	GEN_PMCON1	General PM Configuration 1
0x24	GEN_PMCON2	General PM Configuration 2
0x28	MFPMC	Message from PMC
0x2C	SEC_STS	SEC Status
0x30	CRID	Configured Revision ID
0x34	FUNC_DIS	Function Disable
0x48	ETR	Extended Test Mode Register
0x58	GPIO_ROUT	GPIO_ROUT Register

19.3.2.1 PMC Function Disable Register

The BIOS uses this register to disable a specific function. Upon writing this register, the PMC sets the corresponding Function Disable bit.

Table 19-6. PMC Function Disable Register

Device/Function	Device Name	Register Location
N/A	Reserved	PBASE+0x34[11:0]
Device 20/Function 0	Gigabit Ethernet	PBASE+0x34[12]
Device 20/Function 1	Gigabit Ethernet	PBASE+0x34[13]
Device 20/Function 2	Gigabit Ethernet	PBASE+0x34[14]
Device 20/Function 3	Gigabit Ethernet	PBASE+0x34[15]
N/A	Reserved	PBASE+0x34[16]
Device 23/Function 0	SATA2 Controller with Legacy IDE Mode	PBASE+0x34[17]
Device 22/Function 0	EHCI Controller (USB2)	PBASE+0x34[18]
Device 24/Function 0	SATA3 Controller with Legacy IDE Mode	PBASE+0x34[19]
N/A	Reserved	PBASE+0x34[31:20]

19.3.3 Exiting the G2 (S5) Soft-Off Power State

The sleep state S5 (Soft Off) is exited based on wake events. The wake events force the system to a full-on state (S0); although some non-critical subsystems might still be shutoff and have to be brought back manually. For example, the hard disk may be shutoff during a sleep state and have to be enabled via an I/O pin before it can be used. When exiting from the software-entered sleep states (i.e., those initiated via the PM1_CNT.SLP_EN bit), the PM1_STS_EN.WAK_STS bit is set. After setting the SLP_EN, the operating system polls waiting for the WAK_STS to be set.

The possible causes of wake events (and their restrictions) are shown in Table 19-7.

Table 19-7. Causes of Wake Events

Cause	Well	Type	How Enabled	Wake From S(x)	Wake From Reset Type 5[1]
RTC Alarm	RTC	Internal	Set the RTC_EN bit in the PM1_STS_EN register.	Y	
PMU_PWRBTN_B (Power Button)	SUS	Pin	Always enabled as a wake event	Y	Y
GPIO_SUS0..3	SUS	Pin	GPE0_EN register (after having gone to S5 via SLP_EN but not after a power failure) ***Note:*** GPIOs that are in the core well are not capable of waking the system from sleep states where the core well is not powered.	Y	
PMU_WAKE_B (PCI Express* WAKE#)	SUS	Pin	PCIEXP_WAKE_DIS bit ***Note:*** When the WAKE# pin is active and the PCIEXP_WAKE_DIS bit is clear, the SoC wakes the platform.	Y	
PMU_WAKE_LAN_B (GbE Wake#)	SUS	Internal	Internal signal from GbE to the PMC	Y	
Classic USB	SUS	Internal	Set the USBn_EN bit(s) in the GPE0_EN register.	Y	
Power Management Events	SUS	Internal	PME_B0_EN bit in the GPE0_EN register. This wake status bit includes multiple internal agents: • Integrated LAN • EHCI (USB) • SATA ***Note:*** SATA can only trigger a wake event in S1 (not supported by the SoC), but if it had asserted its PME prior to the S5 entry and the software does not clear PME_B0_STS, a wake event would still result.	Y	
PCI_EXP PME Messages	N/A	N/A	Since the SoC does not support the S1 sleep state, the platform design must use the PCI Express WAKE# pin rather than the messages for wake from S5.		
PMC Initiated	SUS	Internal	No enable bits. The PMC firmware can wake the host independent of the other wake events listed, if desired. A bit is provided in PRSTS for reporting this wake event to the BIOS. ***Note:*** This wake event may be used as a wake trigger on behalf of some other wake source.	Y	Y
Integrated WOL Enable Override	SUS	Internal	WOL Enable Override Bit (in the configuration space)	Y	Y

1. Reset Type 5 is when the state goes straight to S5 and the host stays there. See the description in Section 19.3.1, "Reset Behavior" on page 448.

19.3.4 CPU INIT#, SMI and Reset Generation

See Table 19-8 for the list of registers used to generate resets, a System Management Interrupt (SMI), and the Internal Initialization (INIT#) signal.

Table 19-8. PMC ACPI Registers in Fixed I/O Space

I/O Address (Fixed)	Name	Description
0x00B2	PORTB2	APM Control Register (8-bit read/write scratchpad register). A write also initiates an SMI if enabled.
0x00B3	PORTB3	8-bit read/write scratchpad register.
0x0092	PORT92	Forces the Initialization (INIT#) signal to the CPU.
0x0CF9	RST_CNT	Reset Control register for system and CPU resets.

Ports B2h and B3h are 8-bit, read/write scratchpad registers.

If the APMC_EN bit of the SMI_EN register is set, an I/O write to port B2h also sets the APM_STS bit of the SMI_STS register at offset 34h of base address ABASE in the I/O space and generates an SMI. The SMI_EN is at offset 30h from base address ABASE in the I/O space. See Table 19-9 on page 454.

The following actions send an INIT# signal to the CPU:

- An I/O write to PORT92 where the port INIT_NOW (bit 0) transitions from a 0 to a 1.
- An I/O write to PORTCF9 where the port SYS_RST (bit 1) was a 0 and RST_CPU (bit 2) transitions from 0 to 1.
- A shutdown special cycle from the CPU. Here the INIT# assertion is based on the value of the Shutdown Policy Select (SPS) register.

When the internal INIT# is asserted, it resets the integer registers inside the CPU cores without affecting its internal caches or floating-point registers. The cores then begin execution at the power-on reset vector configured during the power-on configuration.

19.3.5 ACPI Registers

See Table 19-9 for the list of the ACPI registers.

The PMC contains fifteen 32-bit registers in the I/O space it uses for various power management functions and control. The area is located in the I/O space starting at the BAR ACPI Base Address (ABASE). The BAR is located in the configuration space for the ILB PCI device at offset 40h of bus 0, device 31, function 0. Software access to the 128-byte I/O area is controlled by the Enable (EN) bit of the ABASE pointer.

Table 19-9. PMC ACPI Registers in Variable I/O Space

I/O Address Offset from ABASE	Name	Description
0x00	PM1_STS_EN	Power Management 1 Status and enable
0x04	PM1_CNT	Power Management 1 Control
0x08	PM1_TMR	Power Management 1 Timer
0x20	GPE0a_STS	General Purpose Event (GPE) 0 Status
0x28	GPE0a_EN	General Purpose Event (GPE) 0 Enables
0x30	SMI_EN	System Management Interrupt (SMI) Control and Enable
0x34	SMI_STS	SMI Status Register
0x38	ALT_GPIO_SMI	Alternate GPIO SMI Status and Enable Register
0x3C	UPRWC	USB Per-Port Registers Write Control
0x40	GPE_CTRL	General Purpose Event (GPE) Control
0x50	PM2A_CNT_BLK	PM2a Control Block
0x60	TCO_RLD	TCO Reload Register
0x64	TCO_STS	TCO Timer Status
0x68	TCO1_CNT	TCO Timer Control
0x70	TCO_TMR	TCO Timer Register

19.3.6 Legacy Timers

The following legacy timers are supported:

- 24-Bit ACPI Timer – Clocked with a 3.579545-MHz signal derived from the 14.31818-MHz clock. It is enabled by the PMC microprocessor code and is always running. If it expires, an SMI is generated.
- Periodic SMI Timer – Programmed by the operating system to generate an SMI every 8, 16, 32, or 64 seconds depending on the setting of the Period SMI Select (PER_SMI_SEL) bits in the General PM Configuration 2 (GEN_PMCON2) register at PBASE + 0x24 in the memory space.
- Software SMI (SWSMI) Timer – Programmed by the operating system. When it expires, the timer counter stops counting and an SMI is generated.
- TCO Watchdog Timer

19.3.6.1 TCO Watchdog Timer

The software uses the TCO watchdog timer to recover from system-hang situations. The watchdog timer starts after the active-low PMU_PLTRST_B signal deasserts. The software reloads it. A System Management interrupt (SMI) is initiated after the first expiration of the timer if the TCO Enable (TCO_EN) bit of the SMI Control and Enable (SMI_EN) is set at ABASE + 0x30 in the I/O space.

When the timer expires a second time without the software clearing the previous expiration status, the PMC issues a host partition reset. This TCO watchdog timer is disabled through a strapping option and through the No Reboot (NO_REBOOT) bit of the Power Management Configuration (PMC_CFG) register at PBASE + 0x08 in the memory space.

The TCO watchdog timer is halted by setting the TCO Timer Halt (TCO_TMR_HALT) bit of the TCO Timer Control (TCO1_CNT) register at ABASE + 0x68 in the I/O space.

19.3.7 Integrated PMC Microprocessor

Power management is performed primarily by the integrated PMC microprocessor. Its code originates from the platform Flash Memory device that is typically used to store the BIOS. This code is also referred to as the power management firmware. The SoC has secure methods of transferring this code to the PMC microprocessor internal RAM. The RAM is also initialized through various debug tools for troubleshooting.

The PMC microprocessor has code stored in its ROM and is able to function even if its integrated RAM is not loaded.

The SoC has various mechanisms to allow the PMC microprocessor patch code to be authenticated and integrated in the RAM.

When Suspend (SUS) power is active, the internal PMC microprocessor is made active when one of the following happens:

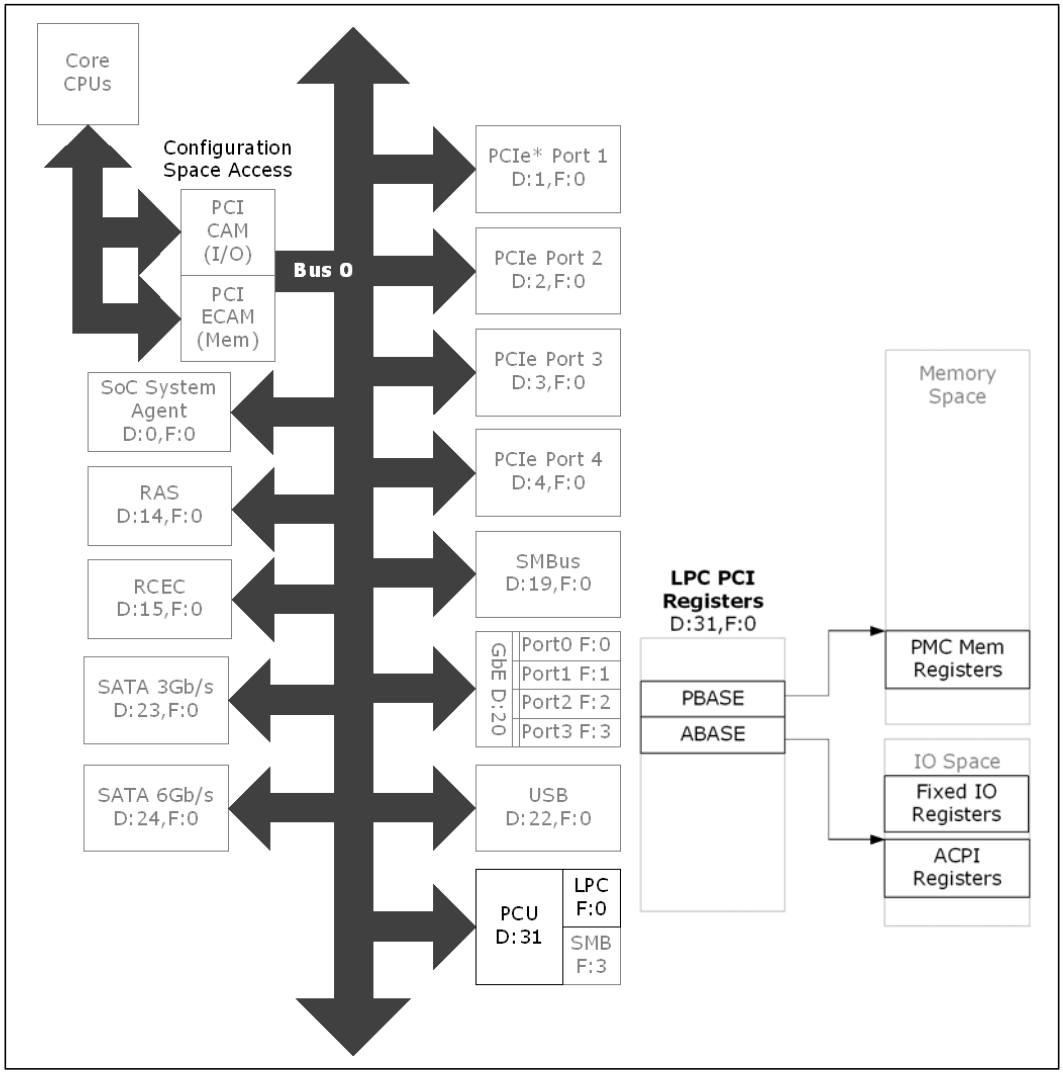
- The power button is pressed.
- An RTC walk-up event occurs.
- The AFTERG3_EN register bit is set in the RTC power well.

The AFTERG3_EN bit, also called AG3E, tells the system whether to boot all the way from the G3 to S0 state or whether to stop in S5 and wait for a wake event before making a transition to the S0 state and booting the system.

19.4 Register Map

Figure 19-2 shows the SoC Power Management Controller registers from a system viewpoint.

Figure 19-2. PMC Register Map

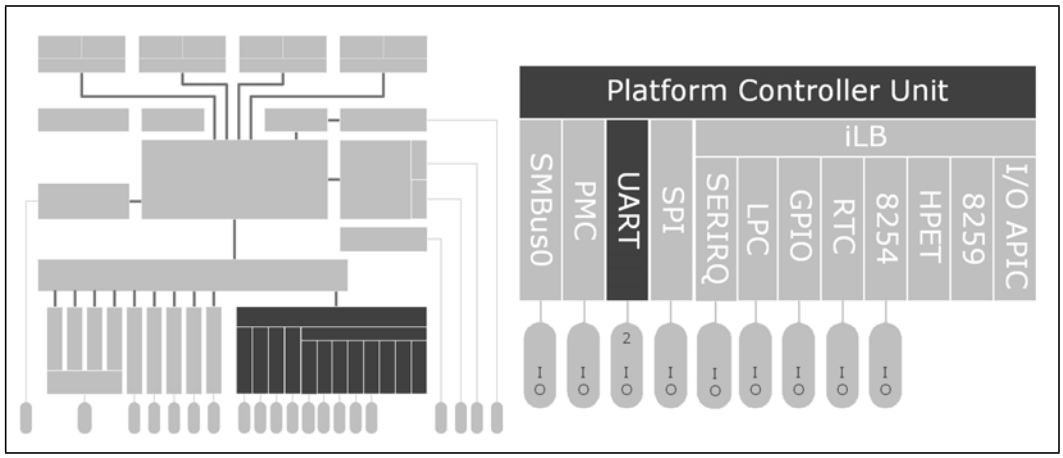


§ §

20 UART Controller

The SoC contains two Universal Asynchronous Receiver/Transmitter (UART) serial ports integrated into the Platform Controller Unit (PCU). The UARTs are controlled by the software using programmed I/O.

Figure 20-1. UART Controller Covered in This Chapter



20.1 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

Table 20-1. Signals

Signal Name	Direction/ Type	Description
UART0_RXD	I muxed	**COM1 Receive:** Data Terminal Equipment (DTE) serial data input from the device pin to the receive port. *This signal is muxed with SMB_DATA2 and GPIOS_13 and is used by other functions.*
UART0_TXD	O muxed	**COM1 Transmit:** DTE serial data output from the transmit port to the device pin. *This signal is muxed with SMB_CLK2 and GPIOS_14 and is used by other functions.*
UART1_RXD	I muxed	**COM2 Receive:** DTE serial data input from the device pin to the receive port. *This signal is muxed with GPIOS_6 and is used by other functions.*
UART1_TXD	O muxed	**COM2 Transmit:** DTE serial data output from the transmit port to the device pin. *This signal is muxed with GPIOS_7 and is used by other functions.*

20.2 Features

Two 16550-compliant UART controllers are available:

- UART0 (COM1)
- UART1 (COM2)

The circuitry is in the core power well.

Each UART interface has 12 registers mapped into the 8-byte addresses in the I/O address space.

- COM1 - 0x3F8-0x3FF
- COM2 - 0x2F8-0x2FF

They use the legacy IRQ#3 and IRQ#4 for interrupts which are sent to the integrated 8259 Programmable Interrupt Controller (PIC).

- COM1 - IRQ4
- COM2 - IRQ3

Only Transmit Data (TXD) and Receive Data (RXD) interface signals are supported.

20.3 Architectural Overview

The UARTs are part of the PCU and are accessed and controlled by the software through the legacy I/O ports in the I/O space. The UARTs do not have any soft straps or straps provided by the platform board hardware. While they are not discovered as a PCI device, the SoC has a register in the configuration space to enable/disable the UART COM interfaces. This register is the UART_CONT register at offset 80h of bus 0, device 31, function 0 in the configuration space. The UART interfaces are enabled by default.

20.4 UART Operation

The serial port consists of a UART which supports a subset of functions of the 16550 industry standard.

The UART performs serial-to-parallel conversion on the data characters received from a peripheral device and parallel-to-serial conversion on the data characters received from the processor. The processor reads the complete status of the UART at any time during the functional operation. Available status information includes the type and condition of the transfer operations being performed by the UART and any error conditions.

The serial port operates in either FIFO or non-FIFO mode. In FIFO mode, a 16-byte transmit FIFO holds data from the processor to be transmitted on the serial link and a 16-byte receive FIFO buffers data from the serial link until read by the processor.

The UART includes a programmable baud rate generator which can generate a baud rate between 50 bps and 115,200 bps from a fixed baud clock input of 1.8432 MHz. The baud rate calculation is:

Equation 20-1. Baud Rate Calculation

$$\text{BaudRate} = \frac{1.8432 \times 10^6}{16 \times \text{Divisor}}$$

The divisor is defined by the Divisor Latch LSB (DLL) and Divisor Latch MSB (DLM) registers of the UART registers in the I/O space. Some common values are shown in Table 20-2.

Table 20-2. Baud Rate Examples

Desired Baud Rate	Divisor	Divisor Latch LSB Register	Divisor Latch MSB Register
115,200	1	1h	0h
57,600	2	2h	0h
38,400	3	3h	0h
19,200	6	6h	0h
9,600	12	Ch	0h
4,800	24	18h	0h
2,400	48	30h	0h
1,200	96	60h	0h
300	384	80h	1h
50	2,304	0h	9h

The UART has interrupt support, and those interrupts are programmed to the user requirements, minimizing the computing required to handle the communications link. Each UART operates in a polled or an interrupt-driven environment as configured by the software.

20.4.1 FIFO Operation

20.4.1.1 FIFO Interrupt Mode Operation

20.4.1.1.1 Receiver Interrupt

When the receive FIFO and receiver interrupts are enabled (FIFO Control Register, bit 0 = 1b and Interrupt Enable Register (IIR), bit 0 = 1b), receiver interrupts occur as follows:

- The receive data available interrupt is invoked when the FIFO has reached its programmed trigger level. The interrupt is cleared when the FIFO drops below the programmed trigger level.
- The IIR receive data available indication also occurs when the FIFO trigger level is reached, and like the interrupt, the bits are cleared when the FIFO drops below the trigger level.
- The receiver line status interrupt (IIR = C6h), as before, has the highest priority. The receiver data available interrupt (IIR = C4h) is lower. The line status interrupt occurs only when the character at the top of the FIFO has errors.
- The Data Ready bit of the Line Status Register (COM[2:1]_LSR.DR) bit is set to 1b as soon as a character is transferred from the shift register to the receive FIFO. This bit is reset to 0b when the FIFO is empty.

20.4.1.1.2 Character Time Out Interrupt

When the receiver FIFO and receiver time out interrupt are enabled, a character time out interrupt occurs when all of the following conditions exist:

- At least one character is in the FIFO.
- The last received character was longer than four continuous character times ago (if two stop bits are programmed, the second one is included in this time delay).
- The most recent processor read of the FIFO was longer than four continuous character times ago.
- The receiver FIFO trigger level is greater than one.

The maximum time between a received character and a time-out interrupt is 160 ms at 300 baud with a 12-bit receive character (i.e., one start, eight data, one parity, and two stop bits).

When a time out interrupt occurs, it is cleared and the timer is reset when the processor reads one character from the receiver FIFO. If a time out interrupt has not occurred, the time out timer is reset after a new character is received or after the processor reads the receiver FIFO.

20.4.1.1.3 Transmit Interrupt

When the transmitter FIFO and transmitter interrupt are enabled (FIFO Control Register, bit 0 = 1b and Interrupt Enable Register, bit 0 = 1b), transmit interrupts occur as follows:

- The transmit data request interrupt occurs when the transmit FIFO is half empty or more than half empty. The interrupt is cleared as soon as the Transmit Holding Register is written (1 to 16 characters are written to the transmit FIFO while servicing the interrupt) or the Interrupt Identification Register is read.

20.4.1.2 FIFO Polled Mode Operation

With the FIFOs enabled (FIFO Control Register, bit 0 = 1b), setting Interrupt Enable Register (IER), bits [3:0] = 000b, puts the serial port in the FIFO polled mode of operation. Since the receiver and the transmitter are controlled separately, either one or both are in the polled mode of operation. In this mode, the software checks the receiver and transmitter status through the Line Status Register (LSR). As stated in the register description:

- LSR[0] is set as long as 1 byte is in the receiver FIFO.
- LSR[1] through LSR[4] specify which error(s) has occurred for the character at the top of the FIFO. The character error status is handled the same way as the interrupt mode. The Interrupt Identification Register is not affected since IER[2] = 0b.
- LSR[5] indicates when the transmitter FIFO needs data.
- LSR[6] indicates that both the transmitter FIFO and shift register are empty.
- LSR[7] indicates whether any errors are in the receiver FIFO.

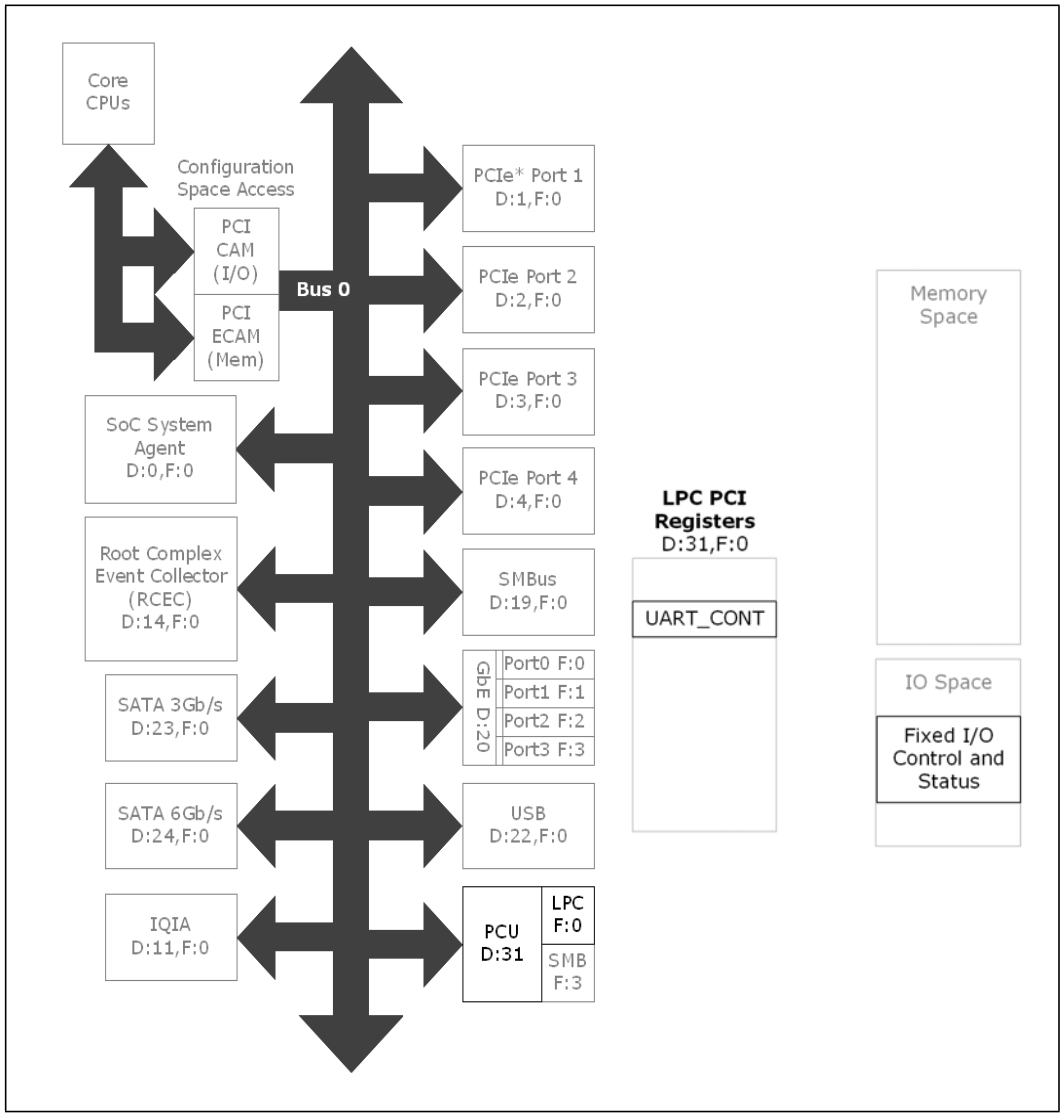
20.5 Registers

The UARTs are enabled/disabled through a register in the configuration space. The UART control and status I/O ports have fixed addresses in the I/O space.

20.5.1 Register Map

Figure 20-2 shows the SoC UART registers from a system viewpoint.

Figure 20-2. UART Registers



20.5.2 PCI Configuration and Capabilities

One 32-bit register, named UART_CONT, is associated with the UARTs and is located in the configuration address space.

Table 20-3. Registers in Configuration Address Space

Configuration Space Address (decimal)	Offset Address	Register Size (bits)	Default	Name	Description
B:0, D:30, F:0	0x80	32	0000_0003h	UART_CONT	UART Control. Used to enable and disable UART ports COM1 and COM2.

Note: Intel recommends the UARTs be disabled during normal platform operation. An enabled UART interferes with platform power management.

20.5.3 Memory-Mapped I/O Registers

No UART registers are located in the memory address space.

20.5.4 Fixed I/O Registers

The following UART registers are located in the I/O address space. All UART I/O registers have fixed addresses defined in the I/O address space.

Table 20-4. Registers in Fixed I/O Address Space

Fixed I/O Address	Name	Description
0x2F8	COM2_Rx_Tx_Buffer	Receiver Buffer/Transmitter Holding Register
0x2F9	COM2_IER	Interrupt Enable Register
0x2FA	COM2_IIR	Interrupt Identification/FIFO Control Register
0x2FB	COM2_LCR	Line Control Register
0x2FC	COM2_MCR	Modem Control Register
0x2FD	COM2_LSR	Line Status Register
0x2FE	COM2_MSR	Modem Status Register
0x2FF	COM2_SCR	Scratchpad Register
0x3F8	COM1_Rx_Tx_Buffer	Receiver Buffer/Transmitter Holding Register
0x3F9	COM1_IER	Interrupt Enable Register
0x3FA	COM1_IIR	Interrupt Identification/FIFO Control Register
0x3FB	COM1_LCR	Line Control Register
0x3FC	COM1_MCR	Modem Control Register
0x3FD	COM1_LSR	Line Status Register
0x3FE	COM1_MSR	Modem Status Register
0x3FF	COM1_SCR	Scratchpad Register

§ §

21 Intel Legacy Block (iLB) Devices

The Intel Legacy Block (iLB) is a collection of disparate functional blocks that are critical for implementing the legacy PC-platform features. It also provides support for Non-Maskable Interrupts (NMI) that are signalled to an open-drain NMI input pin.

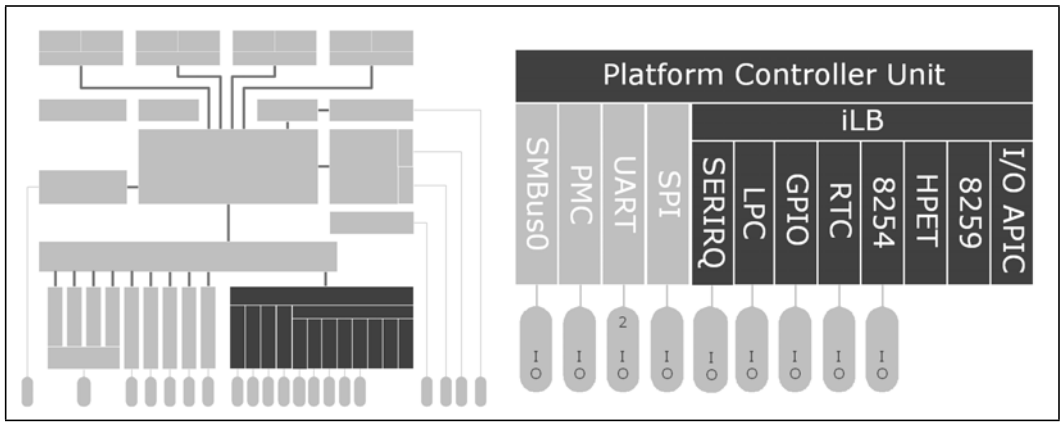
The iLB blocks are:

- Chapter 23, "Serial Interrupt Controller" interface with 31 interrupts synchronized with the LPC clock
- Chapter 24, "Low Pin Count (LPC) Controller" bus host controller and interface
- Chapter 25, "General-Purpose I/O (GPIO)" registers and interface pins
- Chapter 26, "Real Time Clock (RTC)"
- Chapter 27, "8254 Programmable Interval Timer (PIT)" with PC speaker capability
- Chapter 28, "High Precision Event Timer (HPET)"
- Chapter 29, "8259 Programmable Interrupt Controller (PIC)," two cascaded devices
- Chapter 30, "I/O Advanced APIC (I/O APIC)"

The control registers for the iLB blocks are briefly described in this chapter. These legacy-device blocks are described in other chapters.

The PCI-to-ISA bridge and the LPC interface to external devices are described in Chapter 24, "Low Pin Count (LPC) Controller."

Figure 21-1. Intel Legacy Block (iLB) Covered in This Chapter



21.1 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

Table 21-1. Signals

Signal Name	Direction/ Type	Description
NMI	I	**Non-Maskable Interrupt:** This active-high input signal indicates an NMI event. *This signal is muxed with GPIOS_0 and is used by other functions.*

21.2 Features

21.2.1 Key Features

The key features of various iLB blocks are:

- LPC Interface
  - Supports the *Low Pin Count (LPC) 1.1 Specification*
  - No support for DMA or bus mastering
  - Supports Trusted Platform Module (TPM) 1.2
  - Supports keyboard/mouse USB emulation
- General Purpose Input Output
  - Legacy control interface for the SoC GPIOs
  - I/O mapped registers
- 8259 Programmable Interrupt Controller
  - Legacy interrupt support
  - 15 total interrupts through two cascaded controllers
  - I/O mapped registers
- I/O Advanced Programmable Interrupt Controller
  - Legacy-free interrupt support
  - 24 total interrupts
  - Memory-mapped registers
- 8254
  - Legacy timer support
  - Three timers with fixed uses: system timer, refresh request signal and speaker tone
  - I/O mapped registers
- High Performance Event Timers (HPET)
  - Legacy-free timer support
  - Three timers and one counter
  - Memory-mapped registers
- Real-Time Clock (RTC)
  - 242-byte RAM backed by battery (a.k.a. CMOS RAM)
  - Generates a wake/interrupt when time matches the programmed value
  - I/O and indexed registers

21.2.2 Non-Maskable Interrupt (NMI)

Non-Maskable Interrupt (NMI) support is enabled by setting the NMI Enable (NMI_EN) bit, at I/O port 70h, bit 7, to 1b.

NMIs are generated by several sources as described in Table 21-2.

Table 21-2. NMI Sources

NMI Source	NMI Source Enabler/ Disabler	NMI Source Status	Alternate Configuration
SERR# goes active. ***Note:*** An SERR# is only generated internally in the SoC.	NSC.SNE	NSC.SNS	All NMI sources alternatively generate an SMI by setting GNMI.NMI2SMIEN=1b. The SoC uses GNMI.NMI2SMIST for observing an SMI status.
IOCHK# goes active. ***Note:*** An IOCHK# is only generated as an SERIRQ# frame.	NSC.INE	NSC.INS	
NMI is generated from the General Purpose I/O (GPIO). ***Note:*** Active is defined as being on the positive or negative edge of the signal using the GNMI.GNMIED register bit.	GNMI.GNMIED	GNMI.GNMIS	
The software sets the GNMI.NMIN register bit.	GNMI.NMIN	GNMI.NMINS	

21.3 Register Map

21.3.1 Memory-Mapped I/O Registers

The iLB MMIO registers are located in the memory space starting at the base address iLB_BASE_ADDRESS (IBASE). The 32-bit IBASE register is one of the LPC configuration registers in the configuration space of bus 0, device 31 (decimal), function 0, offset 0x50.

Table 21-3. iLB MMIO Registers at IBASE (Sheet 1 of 2)

Memory Address Offset	Name	Description
0x00	ACTL	ACTL
0x04	MC	MC
0x08	PIRQA	PIRQA
0x09	PIRQB	PIRQB
0x0A	PIRQC	PIRQC
0x0B	PIRQD	PIRQD
0x0C	PIRQE	PIRQE
0x0D	PIRQF	PIRQF
0x0E	PIRQG	PIRQG
0x0F	PIRQH	PIRQH
0x10	SCNT	SCNT
0x14	KMC	KMC
0x18	FS	FS
0x1C	BC	BC
0x20	IR0	IR0
0x22	IR1	IR1
0x24	IR2	IR2
0x26	IR3	IR3
0x28	IR4	IR4
0x2A	IR5	IR5
0x2C	IR6	IR6
0x2E	IR7	IR7
0x30	IR8	IR8
0x32	IR9	IR9
0x34	IR10	IR10
0x36	IR11	IR11
0x38	IR12	IR12
0x3A	IR13	IR13
0x3C	IR14	IR14
0x3E	IR15	IR15
0x40	IR16	IR16
0x42	IR17	IR17

Table 21-3. iLB MMIO Registers at IBASE (Sheet 2 of 2)

Memory Address Offset	Name	Description
0x44	IR18	IR18
0x46	IR19	IR19
0x48	IR20	IR20
0x4A	IR21	IR21
0x4C	IR22	IR22
0x4E	IR23	IR23
0x50	IR24	IR24
0x52	IR25	IR25
0x54	IR26	IR26
0x56	IR27	IR27
0x58	IR28	IR28
0x5A	IR29	IR29
0x5C	IR30	IR30
0x5E	IR31	IR31
0x60	OIC	OIC
0x64	RC	RC
0x68	RTM	RTM
0x6C	BCS	BCS - BIOS Control Status
0x70	LE	LE
0x80	GNMI	NMI
0x84	LPCC	LPCC
0x88	IRQE	IRQEN

Additional memory-mapped registers for the controllers are in the iLB. They are mentioned in the remaining chapters of this volume of the Datasheet.

21.3.2 USB Port 64/60 Emulation

This BIOS configurable feature enables emulation of I/O ports 64h and 60h allowing for full PS/2 legacy support for USB keyboards and mice. It is also useful in providing USB keyboard and mouse support in operating systems which does not natively support USB.

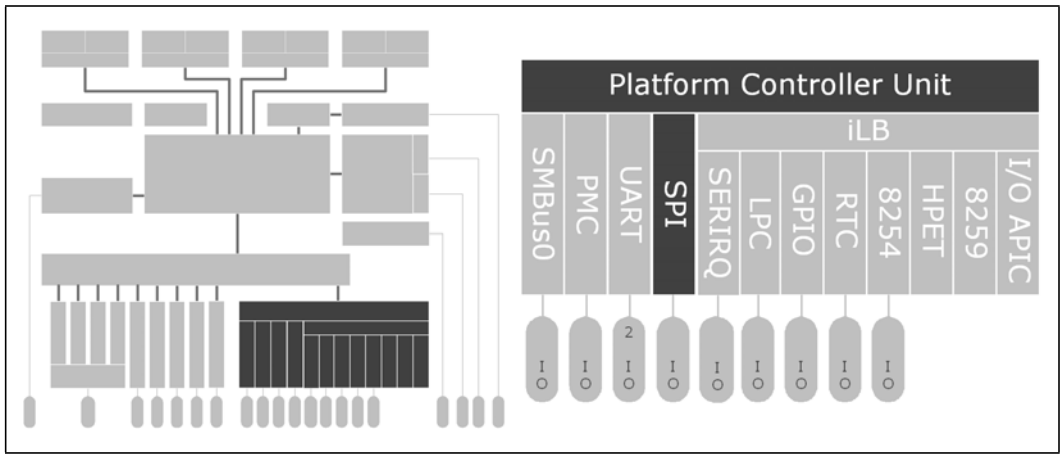
When enabled, the BIOS emulates I/O ports 64h and 60h for the USB keyboard and mouse. This enables PS/2 functionality like keyboard lock, password setting, and scan code selection. When disabled, the BIOS does not emulate I/O ports 64h and 60h for the USB keyboard and mouse.

§ §

22 Serial Peripheral Interface (SPI)

The SoC implements an Serial Peripheral Interface (SPI) controller as one of the two interfaces for the BIOS Flash storage. The controller supports a maximum of two SPI Flash devices and supports frequencies of 20 MHz (default) and 33 MHz.

Figure 22-1. SPI Covered in This Chapter



22.1 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

Table 22-1. SPI Signals

Signal Name	Direction/ Type	Description
SPI_CLK	O	**SPI Clock:** The default is 20 MHz, but can be set to 33 MHz. When the bus is idle, the owner drives the clock signal low.
SPI_CS0_B	O	**SPI Chip Select 0:** Used as the SPI bus request signal for the first SPI Flash device.
SPI_CS1_B	O	**SPI Chip Select 1:** Used as the SPI bus request signal for the second SPI Flash devices. *This signal is muxed with GPIO_SUS12 and is used by other functions.*
SPI_MISO	I	**SPI Master IN Slave OUT:** Data input pin
SPI_MOSI	O	**SPI Master OUT Slave IN:** Data output pin

Table 22-2. SPI Timings - Typical

Parameter	Value	Description
CS# Setup	30 ns (min.)	SPI_CS# low to SPI_CLK high
CS# Hold	30 ns (min.)	SPI_CLK low to SPI_CS# low
Clock High	22 ns (min.)	Time that SPI_CLK is driven high per clock period
Clock Low	22 ns (min.)	Time that SPI_CLK is driven high per clock period

22.2 SPI Features

The following are the SPI features:

- Support for up to two SPI Flash devices.
  - — Storage capacity may be different.
  - — Both devices must be from the same vendor and family.
- Supports five configurable protecting ranges.
- SoC soft-strap information is supported only in the SPI Flash mode.
- Maximum addressability is 16 MB for each SPI device.

All I/O signals are 3.3V and the I/O circuitry is in the Suspend (SUS) power well. The rest of the SPI controller resides in the core power well. During the S5 state, the SPI I/O signals are set as inputs with weak pull-ups to allow the platform board circuitry to access the SPI Flash Memory devices.

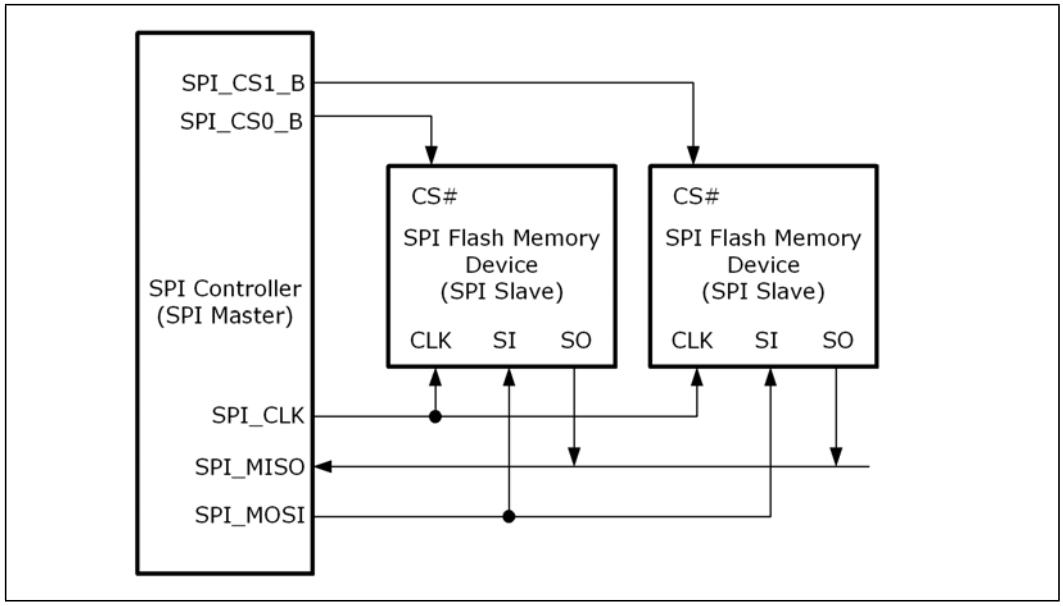
The SPI Cycle Frequency (SCF) register is configured by the BIOS as follows:

- 000: 20-MHz SPI support
- 001: 33-MHz SPI support

22.3 Architectural Overview

Communication on the SPI bus is done with a master–slave protocol. See Figure 22-2 for the master-slave connection of the SPI devices. Communication is full duplex in that data is transferred out at the same time it is transferred into a device. The Slave Output (SO) data is implemented through a tri-state bus. No SPI industry standard exists, but the communication is similar to the SMBus.

Figure 22-2. Connection to the SPI Devices



The SoC can boot the system BIOS and the system firmware through the Low Pin-Count (LPC) bus or through the SPI. The SoC detects from the hard straps which of these two BIOS/firmware boot sources to use. This is described further in Section 16.2, "Pin-Based (Hard) Straps" on page 357.

An SPI Flash Memory device must be connected to Chip Select 0 (the SPI_CS0_B pin/ball) of the SoC and also have a valid descriptor (see Section 22.4, "Operation Modes" on page 474). This is also true in platforms that implement and strap the LPC as the boot source. Here the LPC device contains the boot code, but the soft-strap information is on the SPI memory device.

22.4 Operation Modes

The SoC SPI controller can operate in two different operation modes:

- Descriptor Mode
- Non-Descriptor Mode

The non-descriptor mode is not supported and a valid Flash Descriptor is required for this SoC.

22.4.1 Non-Descriptor Mode

If no valid signature is read (either because no SPI Flash exists, or an SPI Flash exists with no valid descriptor), the Flash controller operates in a non-descriptor mode. This is sometimes referred to as ICH7 mode.

The SoC SPI controller operates in the non-descriptor mode when the contents of the Flash Valid Signature are read and determined to be invalid. This happens if no SPI Flash exists or an SPI Flash exists with no valid descriptor. Also, this holds true regardless if the SPI is configured to be the location of the boot device or if the LPC interface is configured to be the location of the boot device.

The location of the boot device is determined by the Boot BIOS Straps (BBS) field of the General Control and Status register (RCRB_GENERAL_CONTROL). This register is located in the memory space at RCRB_BASE_ADDRESS, offset 0x00. The RCRB_BASE_ADDRESS is in the configuration space at bus 0, device 31 (decimal), function 0, offset 0xF0.

The following features are not supported in the non-descriptor mode:

- Secure Boot
- Soft straps
- Two SPI Flash device support
- Hardware sequencing access
- Descriptor-based security access restrictions

In this mode, software sequencing must be used to access the Flash.

If a Flash Memory device is attached to the SPI controller and the controller is operating in the non-descriptor mode, ensure that the Flash Valid Signature, at offset 10h of the Flash Descriptor, does not equal the expected valid value of 0FF0_A55Ah. Here the SPI controller wrongly interprets that it has a valid signature and that a Flash Descriptor has been implemented.

The SPI allows high-speed support or NOR Flash memory access.

22.4.2 Descriptor Mode

The descriptor mode is required to enable many features:

- Secure Boot
- PCI Express* Root Port configuration
- Support for two SPI components using two separate chip-select pins
- Hardware-enforced security restricting master accesses to different regions
- Soft-strap region providing the ability to use Flash Non-Volatile Memory (NVM) to remove the need for pull-up/pull-down resistors for hard-strapping SoC features
- Support for the SPI fast read instruction and frequencies greater than 20 MHz
- Support for single-input, dual-output fast reads
- Use of standardized Flash instruction set

22.4.2.1 SPI Flash Regions

In the descriptor mode, the Flash is divided into five separate regions as shown in Table 22-3.

Table 22-3. SPI Flash Regions

Region	Content
0	Flash Descriptor
1	BIOS
2	Reserved
3	Reserved
4	Reserved

Only the CPU core running the BIOS code accesses the SPI Flash regions. The only required region is region 0, the Flash Descriptor. Region 0 must be located in the first sector of device 0.

22.4.2.2 Flash Regions Sizes

The SPI Flash space requirements differ by platform and configuration. Table 22-4 indicates the amount of memory needed in the Flash device for each region.

Table 22-4. Region Size Versus Erase Granularity of Flash Components

Region	Size with 4-KB Erase Blocks	Size with 8-KB Erase Blocks	Size with 64-KB Erase Blocks
Flash Descriptor Region	4 KB	4 KB	4 KB
BIOS Region	Varies by Platform	Varies by Platform	Varies by Platform

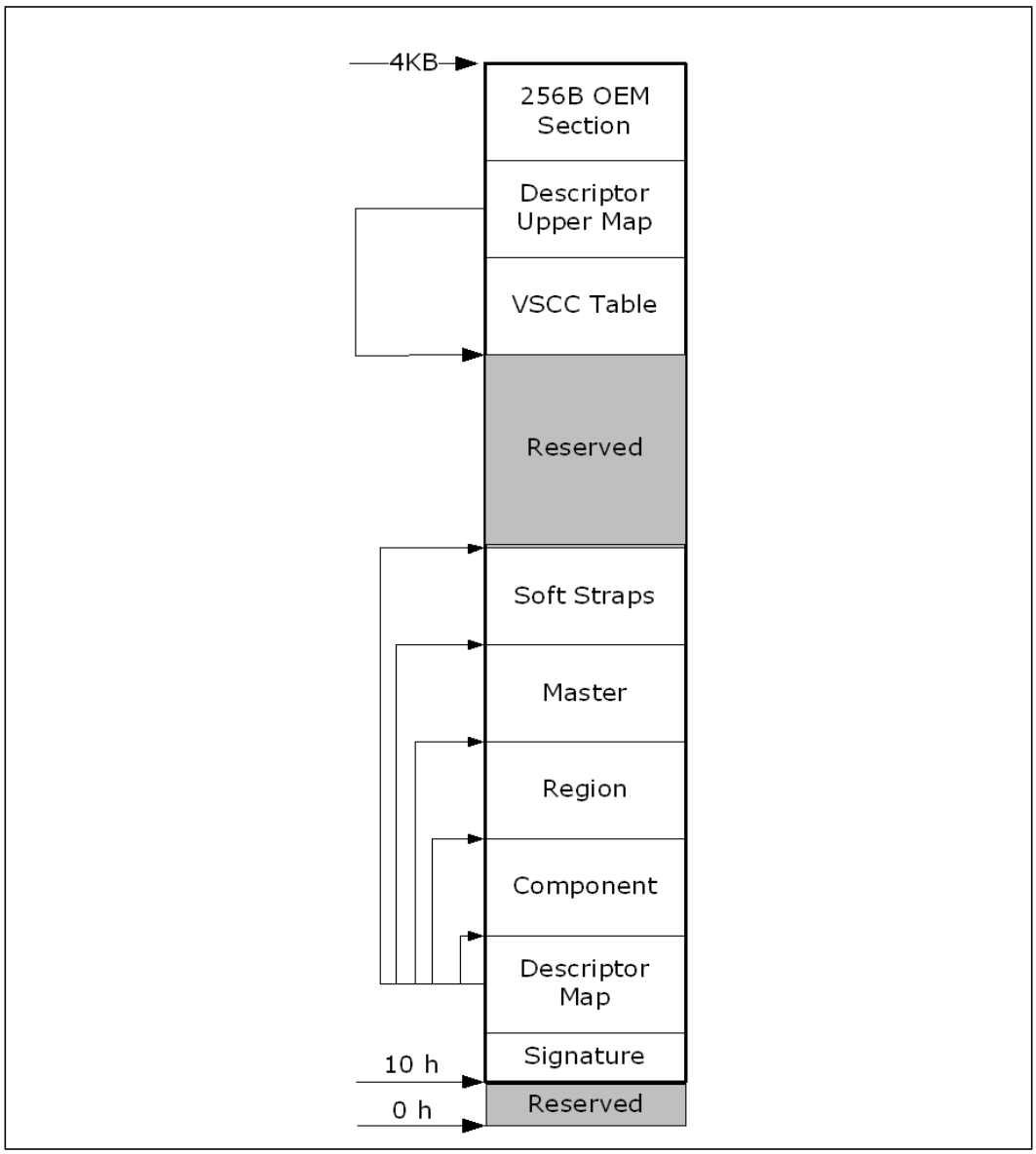
22.5 Flash Descriptor

The maximum size of the Flash Descriptor is 4 KB. If the block/sector size of the SPI Flash device is greater than 4 KB, the Flash Descriptor only uses the first 4 KB of the first block. The Flash Descriptor requires its own block at the bottom of memory (00h).

The information stored in the Flash Descriptor is only written during the board-level manufacturing process. The read/write permissions are set to read only when the system containing the SoC leaves the manufacturing floor.

The Flash Descriptor is made up of eleven sections as indicated in Figure 22-3.

Figure 22-3. Flash Descriptor Sections



- The **Reserved** section at offset 0h is related to functionality not supported.
- The **Signature** section selects the descriptor mode and verifies if the Flash is programmed and functioning. The data at the bottom of the Flash (offset 10h) must be 0FF0A55Ah to be in the descriptor mode.
- The **Descriptor Map** section defines the logical structure of the Flash in addition to the number of components used.
- The **Component** section has information about the SPI Flash in the system including:
  - Density of each component
  - Illegal instructions (such as chip erase)
  - Frequencies for read, fast read, and write/erase instructions
- The **Region** section points to the four other regions and the size of each region.
- The **Master** region contains the security settings for the Flash, grants read/write permissions for each region, and identifies each master by a requestor ID.
- The **Soft Straps** section contains parameter bits that are used to configure the SoC features and/or behaviors. See Section 16.4, “Soft Straps” on page 362.
- The **Reserved** section between the top of the **Soft Straps** section and the bottom of the **VSCC Table** is reserved.
- The **VSCC Table** section holds the JEDEC ID and the Vendor Specific Component Capabilities (VSCC) information of the entire SPI Flash supported by the NVM image.
- The **Descriptor Upper Map** section determines the length and base address of the **VSCC Table** section.
- The **OEM Section** is 256-bytes reserved at the top of the Flash Descriptor for use by an OEM.

22.5.1 Master Section

The master section defines the read and write access setting for each region of the SPI device when the SPI controller is running in a descriptor mode. The master region recognizes only one master: the CPU core running the BIOS code.

22.5.2 Invalid Flash Descriptor Handling

The SoC responds to an invalid Flash Descriptor with the following:

- The SPI controller operates in the non-descriptor mode.
- If the BBS strap is set to 1, the BIOS direct read access is forwarded to the SPI controller without any address translation. See Section 16.2, "Pin-Based (Hard) Straps" on page 357 for BBS details.
- The Flash Descriptor Valid (FDV) bit of the Hardware Sequencing Flash Status (HSFSTS) register bits remain at 0b. The HSFSTS register is located in the MMIO, offset 4, of the SPI controller.
- All security checks are disabled and the entire Flash is open for reading and writing. No restriction is on the 4k crossing.

Note: To ensure BIOS boot access even when the Flash Descriptor is invalid, the BIOS region is placed at the top of Flash component 0. Placing the BIOS region in any other location necessitates a full reprogramming of the Flash before a boot occurs from that Flash.

22.5.3 Descriptor Security Override Strap

A strap is implemented on the UART1_TXD/GPIOS_7 pin (AH50) to allow the descriptor security to be overridden when the strap is sampled low.

If the strap is set (0b), it has the following effect:

- The master region read access and master region write access permissions that were loaded from the Flash Descriptor master section are overridden, giving every master read and write permissions to the entire Flash component including areas outside the defined regions.
- The BIOS Protected Range 4 (PR4), if enabled by a soft strap, is overridden so that all masters are able to write to the PR4. The PR4 base and limit addresses are fetched and received from a soft strap.

22.6 Flash Access

The two types of Flash accesses are:

- Direct Access
- Program Register Access

22.6.1 Direct Access

- Direct writes to the SPI Flash are not allowed by any SoC internal initiator.
- The CPU core is only allowed to do a direct read of the BIOS region.

Note: The BIOS Decode Enable (BDE) register which is located in the configuration space at bus 0, device 31 (decimal), function 0, at offset D8h allows for BIOS MMIO space up to 16MB to be forwarded to the SPI controller. The SPI controller will direct these access to the BIOS region (assuming it is 16MB). All other access must use Program Register Access (see HSFSTS, HSFCTL, FADDR and FDATA0)

22.6.1.1 Security

- The calculated Flash Linear Address (FLA) must fall between the primary region base/limit.

Note: During the non-descriptor mode, the Flash physical address is used instead. Only the two BIOS ranges at the E0000h and F0000h segments below 1 MB are supported.

- Direct read cache contents are reset to 0s on a read from a different master.

22.6.2 Program Register Access

- Reads, writes, and erases are all supported.
- Program register access uses hardware or software sequencing. See Section 22.10, "Hardware vs. Software Sequencing" on page 486 for further information.
- Program register accesses are not allowed to cross a 4-KB boundary and do not issue a command that extends across two components.
- The software programs the FLA corresponding to the region desired.
  - The software must read the devices primary region base/limit address to create an FLA.

Each internal initiator accesses the Flash through a set of memory-mapped registers that are dedicated to each Flash device.

The software uses two separate control and status registers when using program register access to the Flash. The hardware sequencing control/status registers rely on the hardware to issue the appropriate Flash instructions and atomic sequences. The software sequencer puts control into the hands of the software for what instructions to issue and when.

The goal is to support all Flash components through hardware sequencing. Software sequencing is intended only as a back-up strategy.

Note: Software sequencing is required when operating in a non-descriptor mode.

22.6.2.1 Security

- Only the CPU core running the BIOS accesses the registers.
- Using the protected range registers, the BIOS adds separate read/write protection above that granted in the Flash Descriptor for its own accesses.
  - For example, the BIOS wants to protect different regions of the BIOS from being erased.
  - Ranges extend across region boundaries.

22.7 Serial Flash Device Compatibility Requirements

A variety of Serial Flash devices exist in the market. For a Serial Flash device to be compatible with the SPI bus, it must meet the minimum requirements detailed in the following sections of this document.

Note: Intel has validated the Windbond Electronics SPI device, W25Q64FVSSIG. Ensure all functions supported on this device are supported by the SPI device used in your design.

22.7.1 BIOS SPI Flash Requirements

The SPI Flash device must meet the following minimum requirements when used explicitly for the system BIOS storage:

- The erase size has at least one of the following: 64 Kbytes, 8 Kbytes, 4 Kbytes, or 256 bytes.
- The device must support multiple writes to a page without requiring a preceding erase cycle (refer to Section 22.7.3).
- The Serial Flash device must ignore the upper address bits such that an address of FF_FFFFh aliases to the top of the Flash memory.
- SPI Compatible Mode 0 support (the clock phase is 0 and data is latched on the rising edge of the clock).
- If the device receives a command that is not supported or incomplete (less than 8 bits), the device must complete the cycle gracefully without any impact on the Flash content.
- An Erase command (page, sector, block, chip, etc.) must set all bits inside the designated area (page, sector, block, chip, etc.) to 1 (Fh).
- Status register bit 0 must be set to 1 when a write, erase, or write to a status register is in progress and cleared to 0 when a write or erase is NOT in progress.
- Devices requiring the Write Enable command must automatically clear the write enable latch at the end of data program instructions.
- Byte write must be supported. The flexibility to perform a write between 1 byte to 64 bytes is recommended.
- Hardware sequencing requirements are optional in the BIOS-only platforms.
- SPI Flash devices that do not meet the hardware sequencing command set requirements may work in the BIOS-only platforms using software sequencing.

22.7.2 Hardware Sequencing Requirements

Table 22-5 contains a list of commands and the associated opcodes that an SPI-based Serial Flash device must support to be compatible with hardware sequencing.

Table 22-5. Hardware Sequencing Commands and Opcode Requirements

Commands	Opcode	Notes
Write to Status Register	01h	Writes a byte to the SPI Flash Status register. Enable Write to Status Register command must be run before this command.
Program Data	02h	Single-byte or 64-byte write as determined by the Flash part capabilities and software.
Read Data	03h	
Write Disable	04h	
Read Status	05h	Outputs contents of the SPI Flash Status register.
Write Enable	06h	
Fast Read	0Bh	
Enable Write to Status Register	50h	Enables a bit in the status register to allow an update to the status register.
Erase	Programmable	Uses the value from the LVSCC.LEO register or UVSCC.UEO register depending on the FADDR.FLA and whether it is below or above the FPB.FPBA respectively.

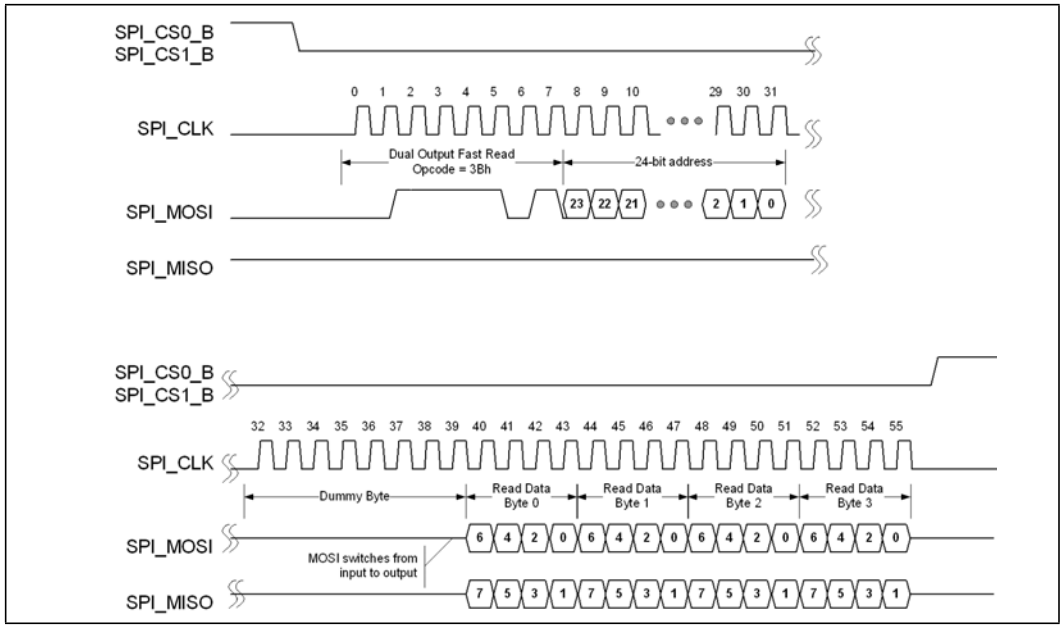
22.7.2.1 Single-Input, Dual-Output Fast Read

The SPI controller supports the functionality of a single-input, dual-output fast read: opcode 3Bh. This instruction has the same timing (including a dummy byte) and the same frequencies as the fast read instruction, with the difference that the read data from the Flash is presented on both the MISO and MOSI pins. During a dual-read instruction, the odd data bits are on the MISO pin and the even data bits are on the MOSI pin.

Note: When the dual-output-fast-read support is enabled the fast-read support must be enabled as well.

Note: The Micronix* SPI Flash uses a different opcode for dual-fast read and requires that during the address phase the address bits are sent on both MOSI and MISO. The SoC does not support this implementation of the protocol.

Figure 22-4. Dual Output Fast Read Timing



22.7.2.2 JEDEC ID

Since each Serial Flash device may have unique capabilities and commands, the JEDEC ID is the necessary mechanism for identifying the device so the uniqueness of the device is comprehended by the controller (master). The JEDEC ID uses the opcode 9Fh and a specified implementation and usage model. This JEDEC Standard Manufacturer and Device ID Read method is defined in Standard JESD21-C, PRN03-NV.

22.7.2.3 Error Correction and Detection

If the first 8 bits specify an opcode which is not supported, the slave does not respond and wait for the next high-to-low transition on PCU_SPI_CS[1:0]#. The SPI controller automatically discards 8-bit words that were not completely received upon deassertion of the signal.

Any other error correction or detection mechanisms must be implemented in the firmware and/or the software.

22.7.3 Multiple Page Write Usage Model

The BIOS usage model requires that the Serial Flash device support multiple writes to a page (minimum of 512 writes) without requiring a preceding Erase command. The BIOS commonly uses capabilities such as counters that are used for error logging and system boot progress logging. These counters are typically implemented by using byte writes to increment the bits within a page that have been designated as the counter. The BIOS multiple page write usage model applies to sequential and non-sequential data writes.

This usage model requirement is based on any given bit only being written once from a 1 to a 0 without requiring the preceding erase. An erase is required to change bits back to the 1 state.

22.8 Soft Flash Protection

Two types of Flash protection are not defined in the Flash Descriptor that are supported by the SPI controller:

1. Flash Range Read and Write Protection
2. Global Write Protection

22.8.1 Flash Range Read and Write Protection

The SPI controller provides a method for blocking reads and writes to specific ranges in the Flash when the protected ranges are enabled. This is achieved by checking the read or write cycle type and the address of the requested command against the base and limit fields of a read or write protected range. The protected range registers are only applied to the programmed register accesses and have no effect on direct reads.

Note: Once the BIOS has locked down the Protected BIOS Range registers, this mechanism remains in place until the next system reset.

22.8.2 Global Write Protection

The SPI controller has a Write Protection Disable (BCR.WPD) configuration bit. When BCR.WPD=0b, the BIOS is not able to perform any Write or Erase commands to the Flash. When BCR.WPD=1b, protection against the BIOS erase and rewrite is disabled. When the Lock Enable (BCR.LE) bit is set, the BIOS disables this protection only during the System Management Mode (SMM) execution.

If BCR.LE=1b, the SPI controller confirms that only SMM code succeeds to set BCR.WPD=1b. In addition, if SCS.SMIWPEN=1b, the SPI controller initiates an SMI when non-SMM code sets BCR.WPD=1b.

22.9 SPI Flash Device Recommended Pinout

This information is in the *Intel® Atom™ Processor C2000 Product Family Platform Design Guide (PDG)*.

22.10 Hardware vs. Software Sequencing

Hardware and software sequencing are the two methods the SoC uses to communicate with the Flash via the programming registers.

22.10.1 Hardware Sequencing

Hardware sequencing has a predefined list of opcodes, see Table 22-5, "Hardware Sequencing Commands and Opcode Requirements" on page 482 for more details, with only the erase opcode being programmable. This mode is only available if the descriptor is present and valid.

22.10.2 Software Sequencing

All commands other than the standard (memory) reads must be programmed by the software in the Software Sequencing Control, Flash Address, Flash Data, and Opcode Configuration registers. The software must issue either Read ID or Read JEDEC ID, or a combination of the two to determine what Flash component is attached. Based on the Read ID, the software determines the appropriate opcode instruction sets to set in the program registers and at what SPI frequency to run the command.

The software must program the flash linear address for all commands, even for those commands that do not require an address such as the Read ID or Read Status. This is because the SPI controller uses the address to determine which chip select to use.

The opcode type and data byte count fields determine how many clocks to run before deasserting the chip enable. The Flash data is always shifted in for the number of bytes specified and the Flash data out is always shifted out for the number of data bytes specified.

Note: The hardware restricts the burst lengths that are allowed.

A status bit indicates when the cycle has completed on the SPI port allowing the host to know when read results are checked and/or when to initiate a new command.

The controller also provides the atomic cycle sequence for performing erases and writes to the SPI Flash. When this bit is 1 (and the Go bit is written to 1), a sequence of cycles is performed on the SPI interface without allowing other SPI devices to arbitrate and interleave cycles to the Flash device. In this case, the specified cycle is preceded by the Prefix command (8-bit programmable opcode) and followed by repeated reads to the Status Register (opcode 05h) until bit 0 indicates the cycle has completed. The hardware does not attempt to check that the programmed cycle is a write or erase.

If a programmed access is initiated (Cycle Go written to 1) while the SPI controller is already busy with a direct memory read, then the SPI host hardware holds the new programmed access pending until the preceding SPI access completes.

Once the SPI controller has committed to running a programmed access, subsequent writes to the programmed cycle registers that occur before it has completed do not modify the original transaction and result in the assertion of the FCERR bit. The software never purposely behaves in this way and relies on this behavior. However, the FCERR bit provides basic error reporting in this situation. Writes to the following registers cause the FCERR bit assertion in this situation:

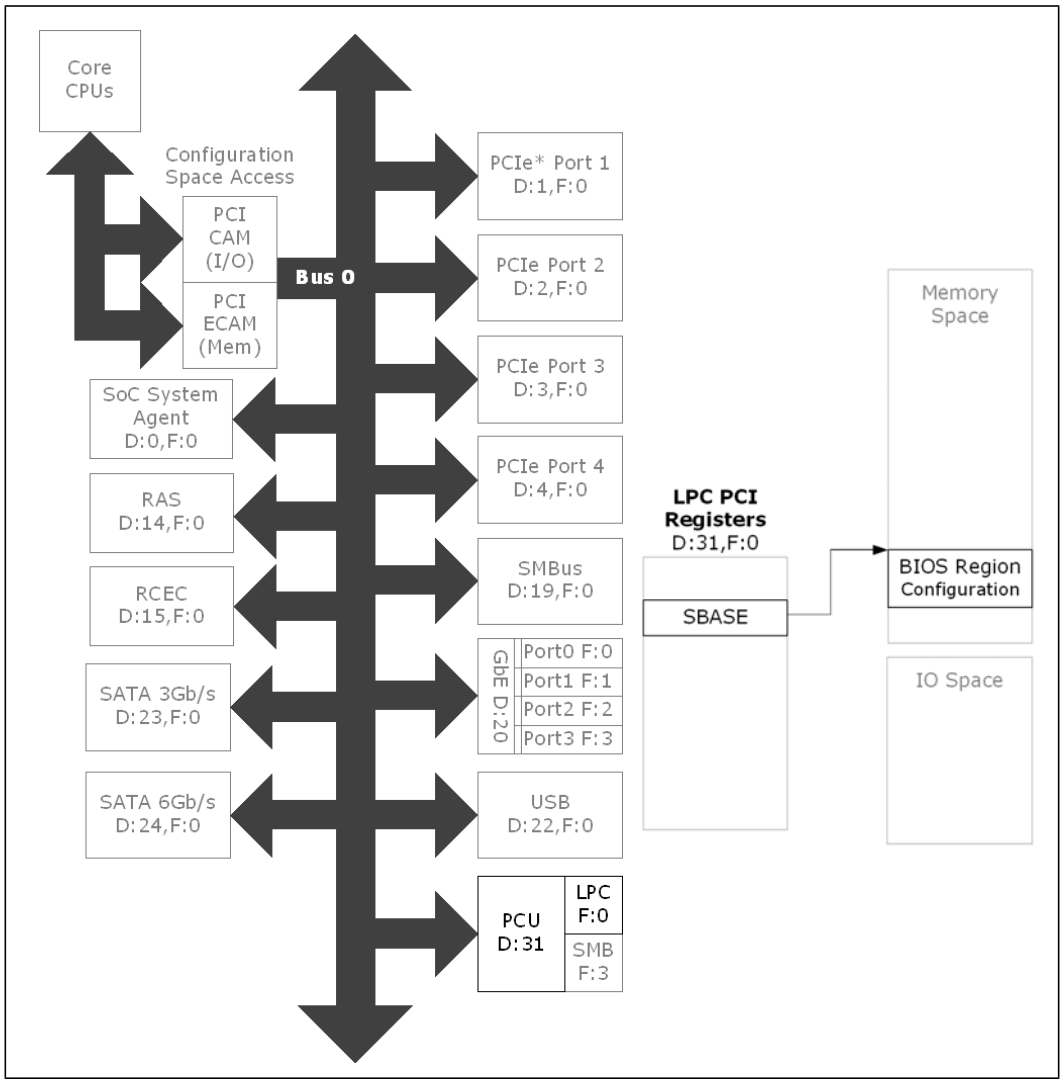
- Software Sequencing Control register
- Software Sequencing Address register
- SPI Data register

With the exception of illegal opcodes, the SPI controller does not police which opcodes are valid to be used in software sequencing. For example, if the software programs a dual-output fast read opcode, then the dual-output fast read cycle is issued, independent of whether the Dual-Output Fast Read Enable bit was set in the component descriptor section.

22.11 Register Map

Figure 22-5 shows the SoC SPI registers from a system viewpoint.

Figure 22-5. SPI Registers



22.11.1 Memory-Mapped Registers

22.11.1.1 BIOS Region (SPI_BIOS_PMA1)

Pointer to base memory address: SPI_BASE_ADDRESS (SBASE)

Pointer location: CFG Offset (0/31/0) 054h

Table 22-6. Map of the BIOS Region (SPI_BIOS_PMA1) Registers (Sheet 1 of 2)

Offset	Name	Description
0x00	BIOS_Flash_Primary_Region_bios	BFPREG
0x04	Hardware_Sequencing_Flash_Status_bios	HSFSTS
0x06	Hardware_Sequencing_Flash_Control_bios	HSFCTL
0x08	Flash_Address_bios	FADDR
0x10	Flash_Data_0_bios	FDATA0
0x14	Flash_Data_1_bios	FDATA1
0x18	Flash_Data_2_bios	FDATA2
0x1C	Flash_Data_3_bios	FDATA3
0x20	Flash_Data_4_bios	FDATA4
0x24	Flash_Data_5_bios	FDATA5
0x28	Flash_Data_6_bios	FDATA6
0x2C	Flash_Data_7_bios	FDATA7
0x30	Flash_Data_8_bios	FDATA8
0x34	Flash_Data_9_bios	FDATA9
0x38	Flash_Data_10_bios	FDATA10
0x3C	Flash_Data_11_bios	FDATA11
0x40	Flash_Data_12_bios	FDATA12
0x44	Flash_Data_13_bios	FDATA13
0x48	Flash_Data_14_bios	FDATA14
0x4C	Flash_Data_15_bios	FDATA15
0x50	Flash_Region_Access_Permissions_bios	FRACC
0x54	Flash_Region_0_bios	FREG0
0x58	Flash_Region_1_bios	FREG1
0x5C	Flash_Region_2_bios	FREG2
0x60	Flash_Region_3_bios	FREG3
0x64	Flash_Region_4_bios	FREG4
0x74	Protected_Range_0_bios	PR0
0x78	Protected_Range_1_bios	PR1
0x7C	Protected_Range_2_bios	PR2
0x80	Protected_Range_3_bios	PR3
0x84	Protected_Range_4_bios	PR4
0x90	Software_Sequencing_Flash_Control_Status_bios	SSFCTLSTS
0x94	Prefix_Opcode_Configuration_bios	PREOP
0x96	Opcode_Type_Configuration_bios	OPTYPE

Table 22-6. Map of the BIOS Region (SPI_BIOS_PMA1) Registers (Sheet 2 of 2)

Offset	Name	Description
0x98	Opcode_Menu_Configuration_0_bios	OPMENU0
0x9C	Opcode_Menu_Configuration_1_bios	OPMENU1
0xB0	Flash_Descriptor_Observability_Control_bios	FDOC
0xB4	Flash_Descriptor_Observability_Data_bios	FDOD
0xC0	Additional_Flash_Control_bios	AFC
0xC4	Lower_Vendor_Specific_Component_Capabilities_bios	LVSCC
0xC8	Upper_Vendor_Specific_Component_Capabilities_bios	UVSCC
0xD0	Flash_Partition_Boundary_bios	FPB
0xF8	SMI_Control_Status_Register_bios	SCS
0xFC	BIOS_Control_Register_bios	BCR
0x100	Trunk_Clock_Gating_Control_bios	TCGC

§ §

23 Serial Interrupt Controller

The SoC does not provide signal pins for the external IRQ signals. For peripherals that need interrupt support, the SoC provides a Serialized Interrupt (SERIRQ) interface through one pin.

The SERIRQ interface accommodates up to with 21 interrupts synchronized with the Low Pin Count (LPC) clocks LPC_CLKOUT[1:0].

Figure 23-1. Serial Interrupt Controller Covered in This Chapter

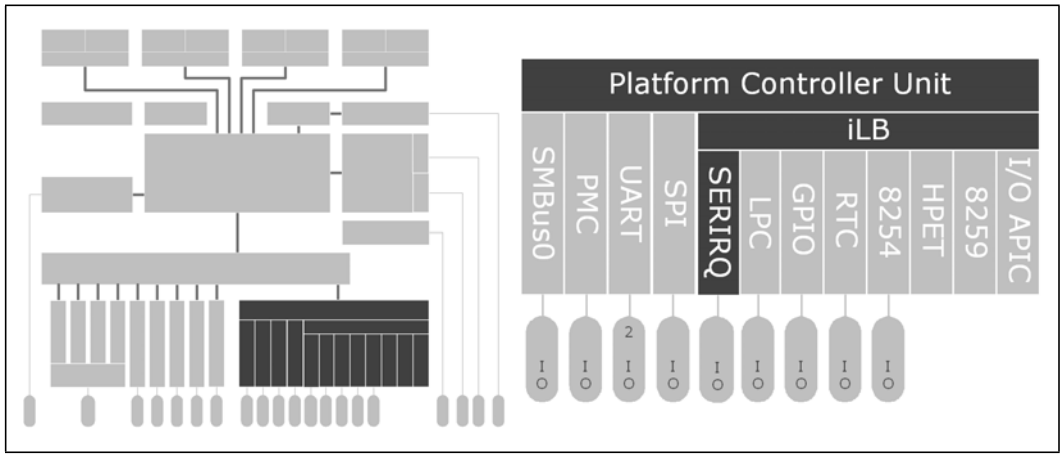


Table 23-1. References

Reference	Revision	Date	Document Title
LPC Specification	1.1	Aug. 2000	*Intel Low Pin Count (LPC) Interface Specification, Revision 1.1*
PCI Specification	3.0	Feb. 3, 2004	*PCI Local Bus Specification*, Revision 3.0

23.1 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

Table 23-2. SoC Serial Interrupt Interface Signals

Signal Name	Direction/ Type	Description
LPC_CLKOUT[1:0]	O	**LPC Clock [1:0] Out:** A 25-MHz PCI-like clock driven to the LPC peripherals. *The LPC_CLKOUT[1:0] signals are used by other functions.*
LPC_CLKRUNB	I/OD	**LPC Clock Run:** Input to determine the status of the LPC clock and an open-drain output is used to request starting or speeding up of the LPC clock. This is a sustained tri-state signal used by the central resource to request permission to stop or slow the LPC clock. The central resource maintains the signal in the asserted state when the LPC clock is running and deasserts the signal to request permission to stop or slow the LPC clock. An external pull-up resistor, 20 kΩ suggested, tied to 3.3V is required. *This signal is used by other functions.*
ILB_SERIRQ	I/O	**Serial Interrupt Request:** This signal implements the serialized interrupt protocol. *This signal is muxed with GPIOS[29] and is used by other functions.*

Note: Only 3.3V LPC devices are supported on the LPC interface.

23.2 Architectural Overview

The Serialized Interrupt (SERIRQ) controller is part of the integrated legacy block iLB. The SERIRQ is programmed through the Serial IRQ Control (SCNT) register which is located in the memory space at ILB_BASE_ADDRESS (IBASE) plus an offset of 10h.

IBASE is a memory-address pointer located in the configuration space at bus 0, device 31 (decimal), function 0, offset 50h.

23.2.1 Controller and Protocol Overview

The SERIRQ controller supports a serialized IRQ mechanism developed during the 1990s. One signal line transmits information between the SERIRQ controller and all of the peripherals that support serialized interrupts. This signal line is attached to the SoC through the ILB_SERIRQ pin. The SoC signal is synchronous to the LPC clock and follows the sustained, tri-state protocol that is used by the LPC-bus signals.

The serialized IRQ protocol defines phases of this sustained tri-state signaling as the following:

- Sample (S) Phase - The ILB_SERIRQ signal is driven low.
- Recovery (R) Phase - The ILB_SERIRQ signal is driven high.
- Turn-around (T) Phase - The ILB_SERIRQ signal is released.

The SoC interrupt controller supports 21 serial interrupts. These represent the 15 ISA interrupts (IRQ0 through IRQ1 and IRQ3 through IRQ15), the four PCI interrupts (INTA, B, C, D), and the control signals SMI# and IOCHK#. The serialized interrupt information is transferred using three types of SERIRQ frames:

- Start Frame - The ILB_SERIRQ signal pin driven low by the SoC serialized interrupt controller to indicate the start of the IRQ transmission.
- Data Frames - The serialized IRQ information transmitted by peripherals to the SoC pin. The serialized interrupt controller supports 21 data frames.
- Stop Frame - The ILB_SERIRQ signal pin driven low by the SoC serialized interrupt controller to indicate the end of transmission and the next mode of operation.

23.2.2 Start Frame

The serialized IRQ protocol has two modes of operation which affect the start frame:

- Continuous Mode - The interrupt controller solely generates the start frame.
- Quiet Mode - The peripheral initiates the start frame, and the interrupt controller completes it.

23.2.2.1 Continuous Mode and Quiet Mode

The mode is indicated by the Mode (MD), bit 7, of the Serial IRQ Control (SCNT) register at offset address 10h of the ILB_BASE_ADDRESS of bus 0, device 31, function 0.

When SCNT.MD is set to a 1, the SERIRQ is in continuous mode. When 0, the SERIRQ is in quiet mode. This bit must be set by the software to guarantee that the first action of the SERIRQ is a start frame. The default setting of the MD bit is quiet mode.

These modes are programmed during the stop frame period discussed later in the chapter. The continuous mode must be entered first, to start the first frame. The start frame duration is eight LPC clock periods. This is considered an interrupt polling mode.

In quiet mode, the ILB_SERIRQ line remains inactive (the signal floats) and is pulled-up between the stop frame and start frame until a peripheral drives ILB_SERIRQ low. The SoC serialized interrupt controller senses the line low and drives it low for the remainder of the start frame. Since the first LPC clock of the start frame was driven by the peripheral, the interrupt controller drives ILB_SERIRQ low for 1 LPC clock less than in the continuous mode. This mode of operation allows for lower-power operation.

Note: Refer to the *Intel® Atom™ Processor C2000 Product Family BIOS Writer's Guide (BWG),* Volume 2 of 2 - Section 12.2 and 12.3 for additional IRQ programming details.

23.2.3 Data Frames

Once the start frame has been initiated, the SERIRQ peripherals start counting frames based on the rising edge of ILB_SERIRQ. Each of the IRQ DATA frames consists of exactly three phases. Each phase is one LPC clock period:

- Sample Phase - During this phase, a device drives ILB_SERIRQ low if its corresponding interrupt signal is low. If its corresponding interrupt is high, then the ILB_SERIRQ device tri-states the ILB_SERIRQ line (lets it float). ILB_SERIRQ remains high due to the pull-up resistors required on the platform board.
- Recovery Phase - During this phase, a peripheral device drives ILB_SERIRQ high if it was driven low during the sample phase. If it was not driven during the sample phase, it remains tri-stated in this phase.
- Turn-around Phase - The device tri-states ILB_SERIRQ.

23.2.4 Stop Frame

After the data frames, a stop frame is driven by the serialized interrupt controller. ILB_SERIRQ is driven low for two or three LPC clocks. The number of clocks is determined by the Mode (MD) register bit mentioned earlier (SCNT.MD). The number of clocks determines the next mode, as indicated in Table 23-3.

Table 23-3. SERIRQ, Stop Frame Width to Operation Mode Mapping

Stop Frame Width	Next Mode
Two LPC clocks	Quiet Mode - Indicating that any SERIRQ device may initiate a start frame.
Three LPC clocks	Continuous Mode - Only the interrupt controller may initiate a start frame.

23.2.5 Serial Interrupts Not Supported

Four interrupts are on the serial stream which are not supported by the serialized interrupt controller:

- IRQ0: heartbeat interrupt generated by counter 0 of the internal 8254 Programmable Interval Timer (PIT).
- IRQ8#: the Real Time Clock (RTC) interrupt is only generated internally.
- IRQ13: this interrupt indicates floating point error and is not supported.
- IRQ14: this interrupt is only generated by the Serial ATA (SATA) controller in SATA legacy mode.

The serialized interrupt controller ignores the states of these interrupts if detected in the IRQ data stream.

23.2.6 Data Frame Format and Issues

Table 23-4 shows the format of the data frames. The decoded serial INT[A:D]# values are internally ANDed in the SoC with the corresponding PCI Express* input signals (PIRQ[A:D]#). Therefore, these four interrupts are shared.

The other interrupts decoded via SERIRQ are also ANDed with the corresponding internal interrupts. For example, if the interrupt vector for IRQ10 is set to be used as the System Control Interrupt (SCI) vector, then it is ANDed with the decoded value for IRQ10 from the SERIRQ stream.

Table 23-4. SERIRQ Interrupt Decoding and Mapping

Data Frame #	Interrupt	Clocks Past Start Frame	Comment
1	IRQ0	2	Ignored. This is only generated via the internal 8524 PIT.
2	IRQ1	5	
3	SMI#	8	If sampled low, causes an SMI# and sets the ILB_SMI Status (ILB_SMI_STS) bit in the SMI Status Register (SMI_STS). This read-only bit is a 1 when the ILB logic is requesting an SMI#.
4	IRQ3	11	
5	IRQ4	14	
6	IRQ5	17	
7	IRQ6	20	
8	IRQ7	23	
9	IRQ8	26	Ignored. IRQ8# is only generated internally by the SoC.
10	IRQ9	29	
11	IRQ10	32	
12	IRQ11	35	
13	IRQ12	38	
14	IRQ13	41	Ignored
15	IRQ14	44	Ignored
16	IRQ15	47	
17	IOCHCK#	50	Same as ISA IOCHCK# going active
18	PCI INTA#	53	
19	PCI INTB#	56	
20	PCI INTC#	59	
21	PCI INTD#	62	

23.3 Power Management

23.3.1 Clock Enabling

The LPC clocks are enabled or disabled by setting or clearing, respectively, the LPCC.LPCCLK[1:0]EN bits.

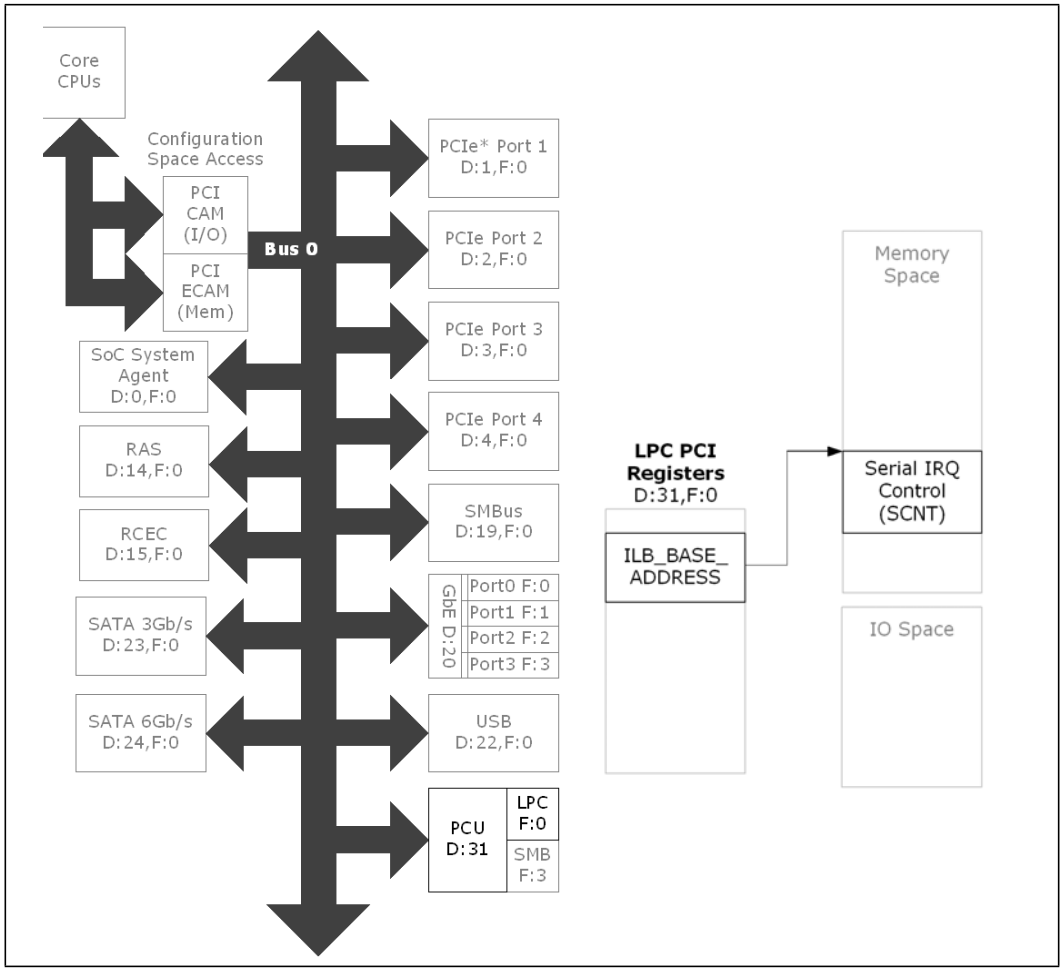
23.3.2 S0idle Support

Details to be provided at a later date.

23.4 Register Map

Figure 23-2 shows the SoC Serialized Interrupt Controller registers from a system viewpoint.

Figure 23-2. SERIRQ Register Map



23.4.1 SERIRQ Registers in Memory Space

The SERIRQ controller is part of the integrated legacy block ILB. The SERIRQ has one 32-bit control register in the memory space at ILB_BASE_ADDRESS (IBASE) plus 10h. IBASE is a memory-address pointer located in the configuration space at bus 0, device 31 (decimal), function 0, offset 50h.

Table 23-5. SERIRQ Register in Memory Space

Offset from IBASE in Memory Space	Name	Description
0x010	SCNT	Serial IRQ Control

§ §

24 Low Pin Count (LPC) Controller

Architecturally, the LPC serves as a PCI-to-ISA bridge to devices connected to the LPC interface pins. The bridge is discovered by the software at bus 0, device 31 (decimal), function 0 in the configuration address space. This bridge also accommodates a number of integrated legacy peripherals, most of which were originally designed for the Industry Standard Architecture (ISA). These legacy peripherals are integrated in the SoC:

- Chapter 23, "Serial Interrupt Controller" interface with 31 interrupts synchronized with the LPC clock
- Chapter 25, "General-Purpose I/O (GPIO)" registers and interface pins
- Chapter 26, "Real Time Clock (RTC)"
- Chapter 27, "8254 Programmable Interval Timer (PIT)" with PC speaker capability
- Chapter 28, "High Precision Event Timer (HPET)"
- Chapter 29, "8259 Programmable Interrupt Controller (PIC)"
- Chapter 30, "I/O Advanced APIC (I/O APIC)"

These legacy devices are described in other chapters. The PCI-to-ISA bridge and the LPC interface to external devices are described in this chapter.

Figure 24-1. LPC Controller Covered in This Chapter

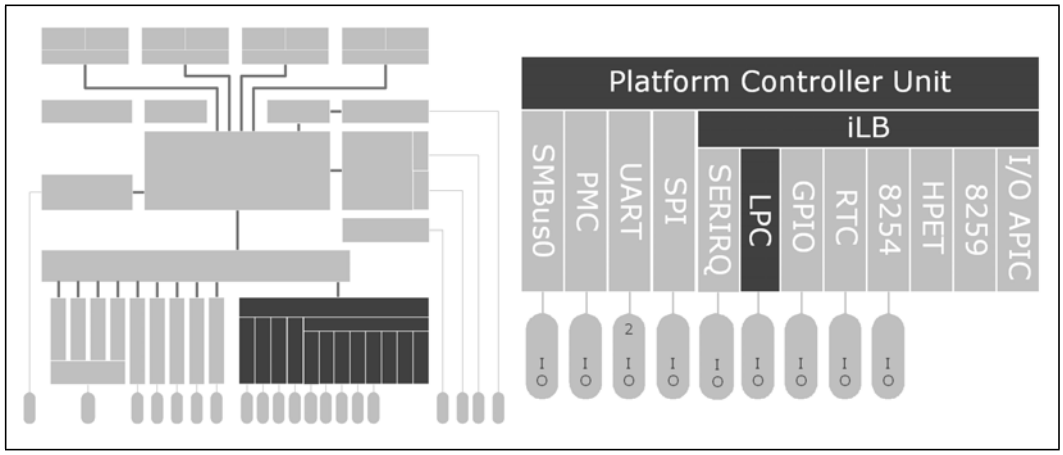


Table 24-1. References

Reference	Revision	Date	Document Title
LPC Specification	1.1	Aug. 2000	*Intel Low Pin Count (LPC) Interface Specification, Revision 1.1*, Revision 1.1
PCI Specification	3.0	Feb. 3, 2004	*PCI Local Bus Specification*, Revision 3.0

24.1 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

Table 24-2. SoC LPC Interface Signals

Signal Name	Direction/ Type	Description
LPC_AD[3:0]	I/O	**LPC Multiplexed Command, Address, Data:** Internal pull-ups are provided for these signals.
LPC_CLKOUT[1:0]	O	**LPC Clock [1:0] Out:** A 25-MHz PCI-like clock driven to LPC peripherals.
LPC_CLKRUNB	I/OD	**LPC Clock Run:** Input to determine the status of LPC_CLK and an open-drain output is used to request starting or speeding up LPC_CLK. This is a sustained tri-state signal used by the central resource to request permission to stop or slow LPC_CLK. The central resource maintains the signal in the asserted state when LPC_CLK is running and deasserts the signal to request permission to stop or slow LPC_CLK. An external pull-up resistor, 20 kΩ suggested, tied to 3.3V is required.
LPC_FRAMEB	O	**LPC Frame:** This signal indicates the start of an LPC cycle or an abort.
ILB_SERIRQ	I/O	**Serial Interrupt Request:** This signal implements the serial interrupt protocol. See Chapter 23, "Serial Interrupt Controller" for additional information about this pin. *This signal is muxed with GPIOS[29] and is used by other functions.*
SUS_STAT_B	O PMU	**SUS_STAT_B**: This active-low output signal indicates that the platform is entering a low-power state (S5) soon. When that happens, the SoC needs to operate from the Suspend (SUS) power well only. This signal is used as the LPC Power Down (**LPCPD#**) signal sent to LPC devices that need the LPCPD# input. **LPCPD#** is monitored by LPC devices with memory that needs to switch from normal refresh mode to suspend refresh mode. It is also used by other peripherals as an indication that they isolate their outputs that are powered by voltage planes that are soon powered-off. This is going to powered-off planes. *This signal is muxed with GPIO_SUS[10] and is used by other functions.*

Note: Only 3.3V LPC devices are supported on the LPC interface.

Table 24-3. LPC Host Signals and the SoC LPC Interface

LPC Specification Signal Name	LPC Peripheral Device Direction	LPC Specification Rev. 1.1 Description	SoC Signal Name (LPC Host)	SoC Direction (LPC Host)
LAD[3:0]	I/O	Multiplexed Command, Address, and Data	LPC_AD[3:0]	I/O
LFRAME#	I	Frame: indicates start of a new cycle and termination of a broken cycle.	LPC_FRAMEB	O
LCLK	I	Clock: same 33-MHz clock as the PCI clock on the host.	LPC_CLKOUT0 LPC_CLKOUT1 Two 25-MHz clock drivers (not 33 MHz) are provided to accommodate the multi-device signal loading.	O This clock is also used internally by the SoC LPC host.

Table 24-3. LPC Host Signals and the SoC LPC Interface

LPC Specification Signal Name	LPC Peripheral Device Direction	LPC Specification Rev. 1.1 Description	SoC Signal Name (LPC Host)	SoC Direction (LPC Host)
CLKRUN#	OD	Clock Run: same as PCI CLKRUN#. Only needed by the peripherals that need DMA or bus mastering in a system that stops the PCI bus (generally in mobile systems).	LPC_CLKRUNB	I/OD
SERIRQ	I/O	Serialized IRQ: only needed by the peripherals that need interrupt support. This signal is required for the host if it does not contain the ISA IRQ lines as inputs.	ILB_SERIRQ This signal is also available to other platform devices to generate serial interrupts to the integrated 8259 PIC.	I/O
LSMI#	OD	SMI#: only needed if a peripheral wants to cause an SMI# on an I/O instruction for retry. Otherwise, use an SMI# via SERIRQ. This signal is optional for the host.	This is connected to any of the SMI-capable GPIO signals of the SoC.	I
LPCPD#	I	Power Down: this indicates that the peripheral prepares for power to be removed from the LPC I/F devices. Actual power removal is system dependent. This signal is optional for the host.	SUS_STAT_B	O
LRESET#	I	Reset: same as PCI reset on the host.	PMU_PLTRST_B	O
LDRQ#	O	Encoded DMA/Bus Master Request: only needed by the peripherals that need DMA or bus mastering.	This is an optional host signal that is not supported by the SoC.	Not applicable
LPME#	OD	LPC Power Management Event: similar to PCI PME#. Used by the peripherals to request wake-up from a low-power state.	This is an optional host signal that is not supported by the SoC.	Not applicable

24.2 Architectural Overview

The LPC serves as a PCI-to-ISA bridge to a number of legacy ISA devices integrated in the SoC and to the external LPC-1.1-compliant devices connected to the LPC interface pins. The bridge is discovered by the software at bus 0, device 31 (decimal), function 0.

The LPC device interface is described in the LPC 1.1 Specification. The specification describes memory, I/O, and DMA transactions. The interface requires a 33-MHz clock that complies with the AC and DC specifications in Section 4.2.2. “3.3V Signaling Environment” of the PCI Local Bus Specification, Revision 3.0. This clock signal is generated by the SoC. Two LPC_CLKOUT clock output balls/pins are provided by the SoC to accommodate multiple LPC devices and independent clock-enable control.

The SoC can boot the system BIOS and the system firmware through the LPC bus or through the Serial Peripheral Interface (SPI). The software selects which of these two BIOS/firmware boot sources to use. The BIOS Soft-Strap register (RCBA+00h[11:10]) must be set to 11b (default) to boot from SPI or 00b to boot from the LPC. This is described further in Chapter 16, "Platform Controller Unit (PCU)."

The LPC interface is used for connection of various legacy components.

24.2.1 No DMA or PHOLD Support

The SoC does not support bus-mastering devices on the LPC interface. Such devices are bi-directional parallel ports, IR controllers, and floppy drive controllers. The LPC does support DMA and what is known as PHOLD, the mechanism for an ISA device to lock the system so that it performs DMA. Super I/O devices that depend on DMA are not supported. If one of these devices is connected to the LPC and a PHOLD is requested, the SoC drops the request, sets an error bit, and the system hangs immediately.

This does not impact other legacy devices such as serial ports, keyboard/mouse, and USB-based peripherals.

24.2.2 LPC Flash Programming Considerations

24.2.2.1 Overview

The Low Pin Count (LPC) interface can be used for connection of various legacy components including: an EC, Super I/O, TPM, FWH. The SoC does NOT support bus-mastering devices on LPC such as Bi-Dir Parallel Port, IR, or Floppy Drive.

This section details BIOS programming considerations for the Firmware Hubs (FWH) located on the LPC.

Note: Only the non-descriptor mode can be used when booting to the BIOS from FWH on the SoC.

The following features are not supported when using LPC (FWH):

- Soft Straps
- Multiple SPI Flash components
- Non-Contiguous Regions
- Direct write of the Flash
- Hardware sequencing cannot be used, software sequencing must be used

For compatibility with the LPC FWH interface, the SPI interface supports decoding the two 64 KB BIOS ranges at the E0000h and F0000h segments just below 1 MB. These ranges must be re-directed (aliased) to the ranges just below 4 GB by the controller. This is done by forcing the upper address bits [23:20] to 1s when performing the read on the SPI interface.

Note: In non-descriptor mode, the SoC will run the BIOS direct read cycle at 20 MHz.

24.2.2.2 Boot BIOS Strap

The SoC can boot from BIOS that resides in Flash, in either the SPI Flash or the LPC (FWH) Flash. There are Boot BIOS System Straps which must be configured to indicate which interface to utilize for BIOS Boot.

Table 24-4. BBS Configurations

Boot BIOS Straps (GCS.BBS)	Description
00	Boot from LPC
11	Boot from SPI
01 or 10	Reserved

Note: The default value of "11b" (Boot from SPI) is set from BIOS soft straps.

Table 24-5. Signal Pin Configurations

SoC Signal Pin Name	Functional Options and Occurrences	SoC Signal Represented by Pin	Direction (I/O)	Internal Pull-up (PU) or Pull-down (PD)	Power Rail
FLEX_CLK_SE0	Strap Sampling	0 = LPC / 1 = SPI	1	20K PU	V3P3S
	As BIOS Starts	FLEX_CLK_SE0	0	None	V3P3S
	SC_USE_SEL = 1	GPIOS_19	Set by SW	TBD	V3P3S

24.2.2.3 LPC Cycle Decoding

Depending on the platform design, the following configuration registers must be programmed properly by System BIOS for SoC to positively decode LPC cycles targeted to the FWH on the LPC:

- Boot BIOS Straps set to "00b" to insure BIOS cycles are routed to LPC
- FWH ID Select registers located at [B:0, D:31, F:0] + 50h are configured
- Enable Bits in the BIOS Decode Enable (BDE) Register are configured

24.2.2.4 LPC Notes

All cycles that are not decoded internally, and are not targeted for LPC (i.e., configuration cycles, IO cycles above 64KB and memory cycles above 16MB), will be sent to LPC with FRAME# not asserted. This aids external debug tools.

24.2.3 Intel® Trusted Platform Module (Intel® TPM)

The LPC interface supports accessing the Intel® Trusted Platform Module (Intel® TPM) 1.2 devices via the LPC TPM START encoding. Software should continue to use the memory mapped 0xFED4_xxxx address range to access the SPI-TPM or the LPC-TPM. No additional checking of the memory cycle is performed.

Since Intel® Trusted Execution Technology (Intel® TXT) transactions are not supported by the SoC, this memory-address range is different than the FED0_0000h to FED4_BFFFh range implemented on some other Intel components.

24.2.4 LPC as the System Subtractive Agent

The LPC Controller is a 32-bit addressed device that sits under the Platform Controller Unit (PCU). These two units make up the System Subtractive Agent for the SoC. This means that a posted or non-posted request targeted for the Memory Mapped I/O (MMIO) or I/O space that is not positively decoded in the SoC will be sent to the PCU/LPC Controller. Here are the rules the SoC follows for all requests that are not positively decoded.

PCU

If an I/O address:

— Forward request to the LPC Controller

If MMIO address < 4 GB:

— Forward request to the LPC Controller

If MMIO address => 4 GB:

— For a read, return an Unsupported Request (UR).

— For a write, it will be silently dropped.

LPC Controller

If an I/O address:

— Place request on the LPC bus (asserts LPC_FRAME_N), if not claimed.

— For a read, return an Unsupported Request (UR).

— For a write, it will be silently dropped.

If MMIO address:

— Place request on the LPC bus, if not claimed.

— For a read, return an Unsupported Request (UR).

— For a write, it will be silently dropped.

Note: To maintain backwards compatibility with older MMIO LPC devices that can only handle addresses < 16 MB, the LPC Controller allows masking the LPC_FRAME_N, if the MMIO address is > 16 MB. See register in Section 24.2.5, “Port 80 POST Code Register Redirection” on page 506.

24.2.5 Port 80 POST Code Register Redirection

The iLB has 16, 1-byte registers accessible in the I/O space at 80h - 8Fh. They are written and read by the software. I/O writes to these locations also pass the write data to the LPC bus for attached POST code displays or indicators typically used for debug purposes. I/O reads by the software to these locations only read the iLB registers and do not result in any LPC transactions.

24.2.6 System Error (SERR)

When an error code (1010) is received on the sync cycle, the LPC controller signals a System Error (SERR) to the core. This occurs only if the SERR reporting is enabled for the LPC.

24.3 Power Management

24.3.1 LPCPD# Protocol

This signal is provided to the LPC peripherals using the SoC output signal SUS_STAT_B. After driving SUS_STAT_B low (active), the SoC drives LPC_FRAMEB (LFRAME#) low, and tri-states or drives low the LPC_AD[3:0] bus.

Note: The Intel Low Pin Count (LPC) Interface Specification, Revision 1.1 defines the LPCPD# protocol where at least 30 µs is from the LPCPD# assertion to the LRESET# assertion. This specification explicitly states that this protocol only applies to entry/exit of low-power states which does not include asynchronous reset events. The SoC asserts both SUS_STAT_B (LPCPD#) and PMU_PLTRST_B (LRESET#) at the same time during a global reset. This is not inconsistent with the LPC LPCPD# protocol.

24.3.2 Clock Run (CLKRUN)

When there are no pending LPC cycles, and the ILB_SERIRQ signal (LPC specification SERIRQ) is in quiet mode, the SoC shuts down the LPC clock. The SoC indicates that the LPC clock is going to shut down by deasserting (driving high) the LPC_CLKRUNB (LPC_CLKRUN#) signal. The LPC devices that require the clock to stay running need to drive LPC_CLKRUNB low within four clocks of its deassertion. If no device drives the signal low within four clocks, the LPC clock stops. If a device asserts LPC_CLKRUNB, the SoC starts the LPC clock and asserts LPC_CLKRUNB.

The CLKRUN protocol is disabled by default and only is enabled during operating system run-time once all LPC devices have been initialized. If the platform board forces the LPC_CLKRUNB signal pin to a continuous logic-low state, the LPC clock is prevented from being shut down by the SoC.

24.3.3 LPC Clock Enabling

The LPC clock signal, LPC_CLKOUT1, is enabled or disabled by the software through the 32-bit LPCC register at offset 84h in the configuration space at bus 0, device 31 (decimal), function 0.

The LPC clock signal, LPC_CLKOUT0, is always enabled. It also has an enable/disable bit in the 32-bit LPCC register at offset 84h, but the bit is read-only and always set to enable LPC_CLKOUT0.

24.4 BIOS and Firmware Flash Memory

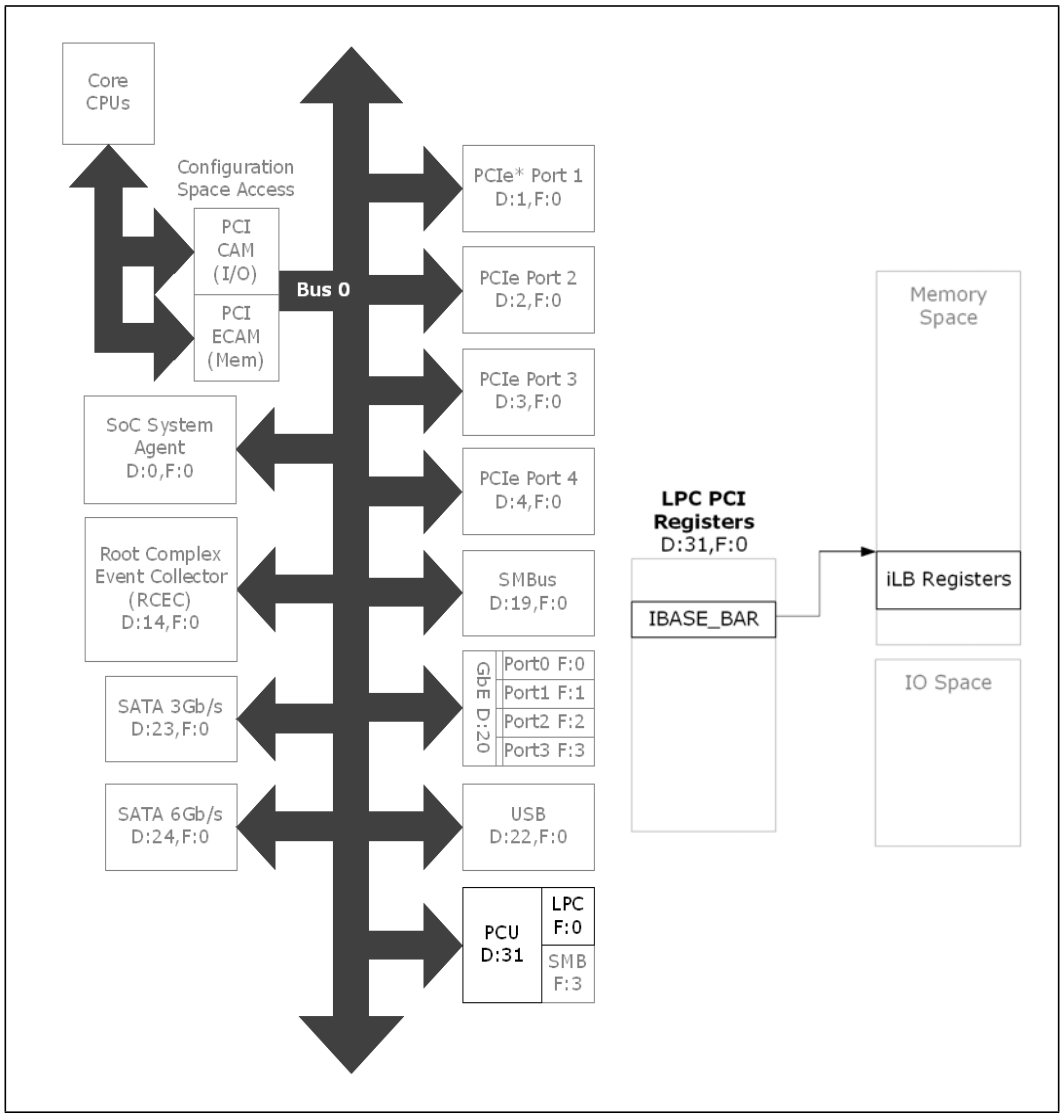
The LPC Firmware Memory Read and Write cycles are intended for the system-boot firmware, although they can be used for any LPC memory cycle. The Sync time depends on the speed of the device. For more information about the Firmware Memory Read and Write cycles, see the "Firmware Memory Cycles" section of the *Intel® Low Pin Count (LPC) Interface Specification, Revision 1.1*. Configuring the SoC to use the LPC for system boot instead of the SPI interface is outlined in Section 24.2, "Architectural Overview" on page 501.

For LPC memory cycles below 16M (100_0000h), the SoC LPC Controller performs standard LPC memory cycles. For cycles targeting firmware (BIOS boot code located at or above 16M), Firmware Memory cycles are used. Only 8-bit transfers are performed. When a larger transfer appears, the LPC controller breaks it into multiple 8-bit transfers until the request is satisfied. If the cycle is not claimed by any peripheral, and subsequently aborted, the LPC controller returns a value of all 1's to the CPU.

24.5 Register Map

Figure 24-2 shows the SoC LPC registers from a system viewpoint.

Figure 24-2. LPC Controller Register Map



24.5.1 PCI Configuration and Capabilities

The LPC device is discovered in the PCI configuration space at bus 0, device 31 (decimal), function 0. The IBASE register pertains to the LPC bus controller as well as various SoC control/status registers. The other registers at bus 0, device 31, function 0 are associated with other integrated devices described in other chapters.

Table 24-6. LPC Register Map - PCI Configuration Space

Configuration Address Offset	Name	Description
0x00	PCIE_REG_Identifiers	Identifiers Register
0x04	PCIE_REG_COMMAND	Command
0x06	PCIE_REG_STATUS	Status
0x08	PCIE_REG_REVISION_ID_CLASS_CODE	Revision ID and Class Code
0x0D	PCIE_REG_MASTER_LAT_TIMER	Master Latency Timer
0x0E	PCIE_REG_HEADER_TYPE	Header Type
0x2C	PCIE_REG_SUBSYS_VENDOR_ID	Subsystem ID and Vendor ID
0x34	PCIE_REG_CAP_POINTER	Capability List Pointer. Points to offset 0xE0.
0x40	ACPI_BASE_ADDRESS	ABASE - ACPI is mapped into 128 bytes of the I/O space.
0x44	PMC_BASE_ADDRESS	PBASE - The PMC registers are mapped into 512 bytes of the memory space.
0x48	GPIO_BASE_ADDRESS	GBASE - The GPIO registers are mapped into 256 bytes of the I/O space.
0x4C	IO_CONTROLLER_BASE_ADDRESS	IOBASE - The I/O controllers registers are mapped into 8 KB of the memory space.
0x50	ILB_BASE_ADDRESS	IBASE - The iLB registers are mapped into 512 bytes of the memory space.
0x54	SPI_BASE_ADDRESS	SBASE - The SPI registers are mapped into 512 bytes of the memory space.
0x58	MPHY_BASE_ADDRESS	MPBASE - M-PHYS registers are mapped into 1 MB of the memory space.
0x5C	PUNIT_BASE_ADDRESS	PUNIT registers are mapped into 2048 bytes of memory space.
0x80	UART_CONT	UART Control - A 32-bit register to enable/disable the UART.
0xD8	PCIE_REG_BIOS_DECODE_EN	BIOS Decode Enable - A 16-bit register that enables ranges in the SPI- or LPC -attached BIOS for decoding purposes.
0xE0	Feature_Detection_Capability_ID	FDCAP
0xE2	Feature_Detection_Capability_Length	FDLEN
0xE3	Feature_Detection_Version_Register	FDVER
0xE4	Feature_Vector_Index	FVECTIDX
0xE8	Feature_Vector_Data	FVECTD
0xF0	RCRB_BASE_ADDRESS	RCBA - The base memory address for the Root Complex registers.

24.5.2 Memory-Mapped I/O Register

The LPC controller has one, 32-bit control register and is called the LPC Control (LPCC) register located in the memory space at the base address ILB_BASE_ADDRESS (IBASE) plus offset 84h.

The 32-bit IBASE register is in the configuration space at bus 0, device 31 (decimal), function 0, offset 0x50.

Table 24-7. Control Register in Memory Space

Memory Address	Name	Description
0x84	LPCC	LPCC

§ §

25 General-Purpose I/O (GPIO)

The SoC provides 59 Customer General-Purpose I/O (GPIO) ports. Each port contains a register that is configured by the platform software, typically the BIOS code, for the customer application. Associated with each port is an external ball/pin with a number of driver/receiver options.

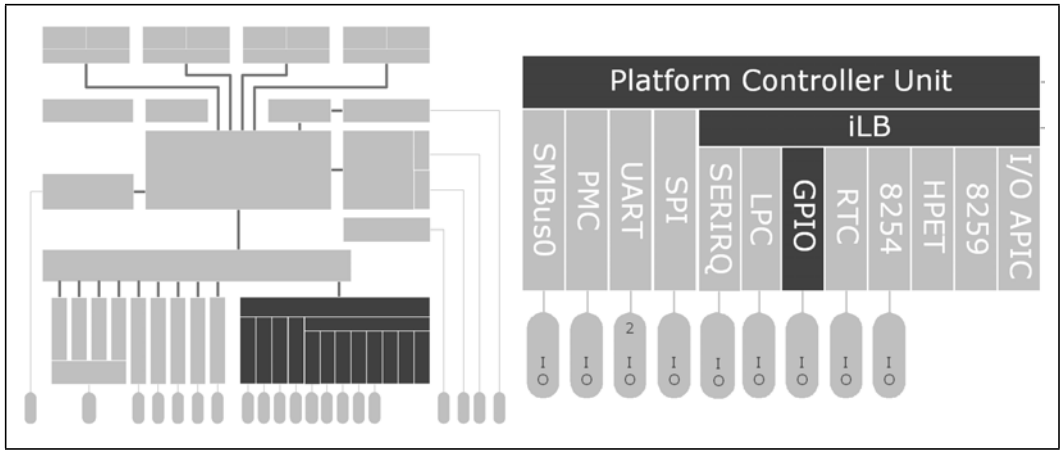
- 31 generic 3.3V GPIOs with circuitry in the core power well
- 28 generic 3.3V GPIOs with circuitry in the Suspend (SUS) power well

Two of the GPIOs in the SUS power well are always available to the customer to use as Customer GPIOs. The other SUS and core Customer GPIOs are designed for multiple uses and may not be available for customer general use if particular system functions are needed. This multi-use feature of the ball/pins is also called GPIO muxing.

The number of the Customer GPIOs that are multi-use depends on the SKU. Customer GPIOs not required by a particular SKU can be configured and used as needed.

Some of the Customer GPIOs have pre-defined characteristics that are used only at reset time. They serve as SoC pin straps that are sampled and retained by the SoC at that time. This document refers to these as the system functional hard pin straps and are described in Table 16-1, "Hard Pin Straps" on page 357. Hard pin straps after reset time, function as the SoC native signals. Once a Customer GPIO is programmed and enabled, the characteristics of the Customer GPIO override the native function of the pin.

Figure 25-1. GPIO Covered in This Chapter



25.1 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

Table 25-1. Signals

Signal Name	Direction/ Type	Description
GPIO_SUS[0] GPIO_SUS[2]	I/O	**GPIO SUS:** These signals are located in the Suspend (SUS) power well and are configured and used by the customer. These signals are NOT shared with the other SoC native functions. Even so, they are hard pin-strap input pins during reset.
GPIO_SUS[1] GPIO_SUS[27:3]	I/O	**GPIO SUS:** These signals are located in the SUS power well and are configured and used by the customer. These GPIOs have multiple uses and their functions depend on how the GPIOs are configured. Some are hard pin-strap input pins during reset.
GPIOS[30:29] GPIOS[20:0]	I/O	**GPIO Core:** These signals are located in the core power well and are configured and used by the customer. These GPIOs have multiple uses and their functions depend on how the GPIOs are configured.
GPIOS[28:21]	I/O	**GPIO Core:** Reserved - These signals are located in the Core power well and are shared with the LPC signals. These signals must be left in their default LPC state and not de-selected via software to be GPIO pins.

The hard pin straps, the native functional signals, and internal termination that are shared with each Customer GPIO pin are shown in Table 31-24, “Signal Pins with Shared Functions - Core Power Well” on page 612 and Table 31-25, “Signal Pins with Shared Functions - SUS Power Well” on page 614.

25.2 Features

- Edge-Detect (E-Detect) capability for predefined pads
  - Wake-up event input pin capability for GPIO_SUS[3:0] and GPIO_SUS[7:6]
  - E-event capture capability for GPIOS[7:0]
- Alternate functions for GPIOs to reduce the SoC pin count
- Filter (de-glitch) capabilities for predefined pads
  - GPIO_SUS[3:0] and GPIO_SUS[9:8]
  - GPIOS[7:0] and GPIOS[18:15]
- Customer GPIO interface, located in the I/O addressing space
- Electrical parameters controlled by the software via registers in the memory space
- Consolidate all DFx and functional muxing at one entity

25.3 Architectural Overview

If their native functions are not used, a number of the SoC signal pins can be programmed to be used as Customer General-Purpose IO pins. Typically this programming is performed by the BIOS. The software uses a number of GPIO configuration registers to make this choice.

Each Customer GPIO port is implemented as dual-read/write registers with its own dedicated storage. From the CPU standpoint, a write value is stored in the write register, while the read value comes from the GPIO read register.

25.3.1 Choosing the Native Signal Mode or Customer GPIO Mode

The individual control and software access to the 31 GPIO ports in the core power well and the 28 GPIO ports in the Suspend (SUS) power well are done through addresses in the I/O space. This addressing starts at the I/O space base address in GPIO_BASE_ADDRESS (GBASE) which is located in the configuration space at bus 0, device 31 (decimal), function 0, offset 0x048.

The 31 GPIO ports in the core well are handled separately from the 28 GPIO ports in the SUS well.

- Six 32-bit GPIO control/access registers (see Table 25-2) are associated with the 31 Customer GPIO ports in the core well. Each bit represents a Customer GPIO register in the core well (bit 0 = GPIOS0, bit 1 = GPIOS1, etc.). Bits 31 and [28:21] are reserved.
- Seven 32-bit GPIO control/access registers (see Table 25-3) are associated with the 28 Customer GPIO ports in the SUS well. Each bit represents a Customer GPIO register in the SUS well (bit 0 = GPIO_SUS0, bit 1 = GPIO_SUS1, etc.). Bits 28 through 31 are reserved.

Table 25-2. GPIO Core Control/Access Registers in I/O Space

Offset from GBASE	Name	Bits Used of the 32-Bit Register	Description
00h	SC_USE_SEL	31 (one per GPIO)	Use Select
04h	SC_IO_SEL	31 (one per GPIO)	Input Output Select
08h	SC_GP_LVL	31 (one per GPIO)	GPIO Level
0Ch	SC_TPE	31 (one per GPIO)	Trigger Positive Edge Enable
10h	SC_TNE	31 (one per GPIO)	Trigger Negative Edge Enable
14h	SC_TS	31 (one per GPIO)	Trigger Status

Note: Bit 0 of each register corresponds to GPIOS0. Bit 1 corresponds to GPIOS1, and so on.

Table 25-3. GPIO SUS Control/Access Registers in I/O Space

Offset from GBASE	Name	Bits Used of the 32-Bit Register	Description
80h	SUS_USE_SEL	28 (one per GPIO)	Use Select
84h	SUS_IO_SEL	28 (one per GPIO)	Input Output Select
88h	SUS_GP_LVL	28 (one per GPIO)	GPIO Level
8Ch	SUS_TPE	28 (one per GPIO)	Trigger Positive Edge Enable
90h	SUS_TNE	28 (one per GPIO)	Trigger Negative Edge Enable
94h	SUS_TS	28 (one per GPIO)	Trigger Status
98h	SUS_WAKE_EN	28 (one per GPIO)	Wake Enable

Note: Bit 0 of each register corresponds to GPIO_SUS0. Bit 1 corresponds to GPIO_SUS1, and so on.

25.3.1.1 SC_USE_SEL and SUS_USE_SEL Registers

- Use as a Customer GPIO
  - When the bit corresponding to a particular Customer GPIO pin is set to a 1 in the USE_SEL register, the pin is available as a general-purpose IO.
- Use as a Native Signal
  - When the bit corresponding to a particular Customer GPIO pin is set to a 0 in the USE_SEL register, the pin is used in its native signal mode. Most pins that can be programmed as Customer GPIOs have only one native signal as an alternative. A small number of Customer GPIOs in the Suspend (SUS) power well have two possible native signals as an alternative.

25.3.2 Electrical Configuration Registers for GPIO Ports

The ball/pin signal I/O buffers are set as 3.3V buffers and cannot be changed.

Besides what can be configured through the USE_SEL registers, some of the other electrical characteristics of the signal balls/pins of the 31 Customer GPIO ports in the core well and the 28 Customer GPIO ports in the SUS well can also be configured by the software. This is accomplished via the registers accessible in the memory space which are addressed as an offset from the IO_CONTROLLER_BASE_ADDRESS (IOBASE) and is located in the configuration space at bus 0, device 31 (decimal), function 0, offset 0x04C.

For additional details reference Section 16.3, “Multi-Functional Signal Pins” on page 360.

Note: Intel has not specified what electrical characteristics can be set by the software for each GPIO pin other than what is possible through the registers offset from GPIO_BASE_ADDRESS (GBASE) in the I/O space. At this time, board designers must not use the registers offset from IO_CONTROLLER_BASE_ADDRESS (IOBASE) in the memory space to configure the Customer GPIO pins.

25.3.3 Using Customer GPIOs in a Board Design

Because most of the GPIO pins are shared with the other SoC functional signals, the board designer must pay attention to the way these pins are configured by the SoC after reset and before the software sets the GPIO USE_SEL bit to 1.

The safest method to avoid circuit contention on the board is to configure a GPIO pin to have the same input/output assignment as the SoC native signal for that pin. The internal termination resistors must also be taken into account. Refer to Table 31-24, “Signal Pins with Shared Functions - Core Power Well” on page 612 and Table 31-25, “Signal Pins with Shared Functions - SUS Power Well” on page 614 for the pin signal direction and internal termination of the shared signals. Also, see Section 31.21, “Signal Pins with Shared Functions or GPIO” on page 612.

The GPIO pins that are also hard-strap pins need this special attention too. When a pin is sampled for the strap value, the SoC treats the pin as an input regardless of its native-signal direction. During this strap-sampling time, a special internal Pull-Up (PU) or Pull-Down (PD) resistor may be tied to the pin regardless of the internal termination of the native function.

Table 25-4 shows the offset of these registers from the IOBASE for each of the 31 core-well GPIO ports. Table 25-5 on page 517 shows these locations for the 28 GPIOs in the SUS power well.

Table 25-4. Customer GPIO Port Configuration Registers - Core Power Well

Customer GPIO when SC_USE_SEL Bit = 1	SoC Ball/Pin Number	SoC Power Well	Bit in GBASE + 00h-17h Registers	Native Signal when SC_USE_SEL Bit = 0
GPIOS_0	AL56	Core	0	NMI
GPIOS_1	AL63	Core	1	ERROR2_B
GPIOS_2	AL62	Core	2	ERROR1_B
GPIOS_3	AL65	Core	3	ERROR0_B
GPIOS_4	AM52	Core	4	IERR_B
GPIOS_5	AL52	Core	5	MCERR_B
GPIOS_6	AG50	Core	6	UART1_RXD
GPIOS_7	AH50	Core	7	UART1_TXD
GPIOS_8	AN62	Core	8	SMB_CLK0
GPIOS_9	AP62	Core	9	SMB_DATA0
GPIOS_10	AL58	Core	10	SMBALRT_N0
GPIOS_11	AN63	Core	11	SMB_DATA1
GPIOS_12	AR63	Core	12	SMB_CLK1 or SPKR
GPIOS_13	AN65	Core	13	SMB_DATA2 or UART0_RXD
GPIOS_14	AR65	Core	14	SMB_CLK2 or UART0_TXD
GPIOS_15	AT63	Core	15	SATA_GP0
GPIOS_16	AL49	Core	16	SATA_LEDN
GPIOS_17	AH51	Core	17	SATA3_GP0
GPIOS_18	AH54	Core	18	SATA3_LEDN
GPIOS_19	AH59	Core	19	FLEX_CLK_SE0
GPIOS_20	AG56	Core	20	FLEX_CLK_SE1
GPIOS_29	AT50	Core	29	ILB_SERIRQ
GPIOS_30	AM58	Core	30	PMU_RESETBUTTON_B

Table 25-5. Customer GPIO Port Configuration Registers - SUS Power Well

Customer GPIO when SUS_USE_SEL Bit = 1	SoC Ball/Pin Number	SoC Power Well	Bit in GBASE + 80h-9Bh Registers	Native Signal when SUS_USE_SEL Bit = 0
GPIO_SUS0	V66	SUS	0	GPIO_SUS0
GPIO_SUS1	W54	SUS	1	GPIO_SUS1 or NCSI_RXD0
GPIO_SUS2	T53	SUS	2	GPIO_SUS2
GPIO_SUS3	Y63	SUS	3	CPU_RESET_B
GPIO_SUS4	Y57	SUS	4	SUSPWRDNACK
GPIO_SUS5	AD58	SUS	5	PMU_SUSCLK
GPIO_SUS6	AC52	SUS	6	PMU_SLP_DDRVTT_B
GPIO_SUS7	Y50	SUS	7	PMU_SLP_LAN_B
GPIO_SUS8	AD66	SUS	8	PMU_WAKE_B
GPIO_SUS9	AC49	SUS	9	PMU_PWRBTN_B
GPIO_SUS10	AB65	SUS	10	SUS_STAT_B
GPIO_SUS11	AD63	SUS	11	USB_OC0_B
GPIO_SUS12	AC58	SUS	12	SPI_CS1_B
GPIO_SUS13	W51	SUS	13	GBE_EE_DI
GPIO_SUS14	W60	SUS	14	GBE_EE_DO
GPIO_SUS15	T50	SUS	15	GBE_EE_SK
GPIO_SUS16	R59	SUS	16	GBE_EE_CS_N
GPIO_SUS17	T58	SUS	17	GBE_SDP0_0
GPIO_SUS18	T48	SUS	18	GBE_SDP0_1 or NCSI_ARB_IN
GPIO_SUS19	P46	SUS	19	GBE_LED0
GPIO_SUS20	W50	SUS	20	GBE_LED1
GPIO_SUS21	P48	SUS	21	GBE_LED2
GPIO_SUS22	R58	SUS	22	GBE_LED3
GPIO_SUS23	V63	SUS	23	NCSI_RXD1
GPIO_SUS24	W56	SUS	24	GBE_MDIO0_I2C_CLK
GPIO_SUS25	W59	SUS	25	GBE_MDIO0_I2C_DATA
GPIO_SUS26	Y54	SUS	26	GBE_MDIO1_I2C_CLK or NCSI_TXD1
GPIO_SUS27	Y53	SUS	27	GBE_MDIO1_I2C_DATA or NCSI_TXD0

25.3.4 GPI-Signaled Events

Not all of the Customer GPIOs have the edge-detect capability. The following do have this capability:

- GPIOS[7:0]
- GPIO_SUS[3:0] and GPIO_SUS[7:6]

When programmed to be used as a Customer GPIO (the SC_USE_SEL/SUS_USE_SEL register bit is 1 for the SC/SUS GPIO), the GPIO supports General-Purpose Input (GPI) edge-triggered events. All GPIO input pins are individually configured by the software to generate a System Management Interrupt (SMI) or System Control Interrupt (SCI). The Customer GPIOs do not have the capabilities to generate IRQ interrupts.

These Customer GPIO input pins are setup to for positive- or negative-edge detection to indicate the event. This is done by programming the GP_TPE and GPE_TNE registers. When the GPI edge event is detected, the SoC sends the appropriate SMI or SCI message to the CPU.

Besides setting the corresponding bit to 1 of the SC_USE_SEL/SUS_USE_SEL register, the polarity of the GPI event is configured in the SC_TPE/SUS_TPE and SC_TNE/SUS_TPE registers.

Note: Refer to the GPE0a_EN - General Purpose Event 0 Enables (GPE0a_EN)—Offset 28h, ALT_GPIO_SMI - Alternate GPIO SMI Status and Enable Register (ALT_GPIO_SMI)—Offset 38h, and GPIO_ROUT - GPIO_ROUT Register (GPIO_ROUT)—Offset 58h registers for information about routing these Customer GPIOs to SMI or SCI.

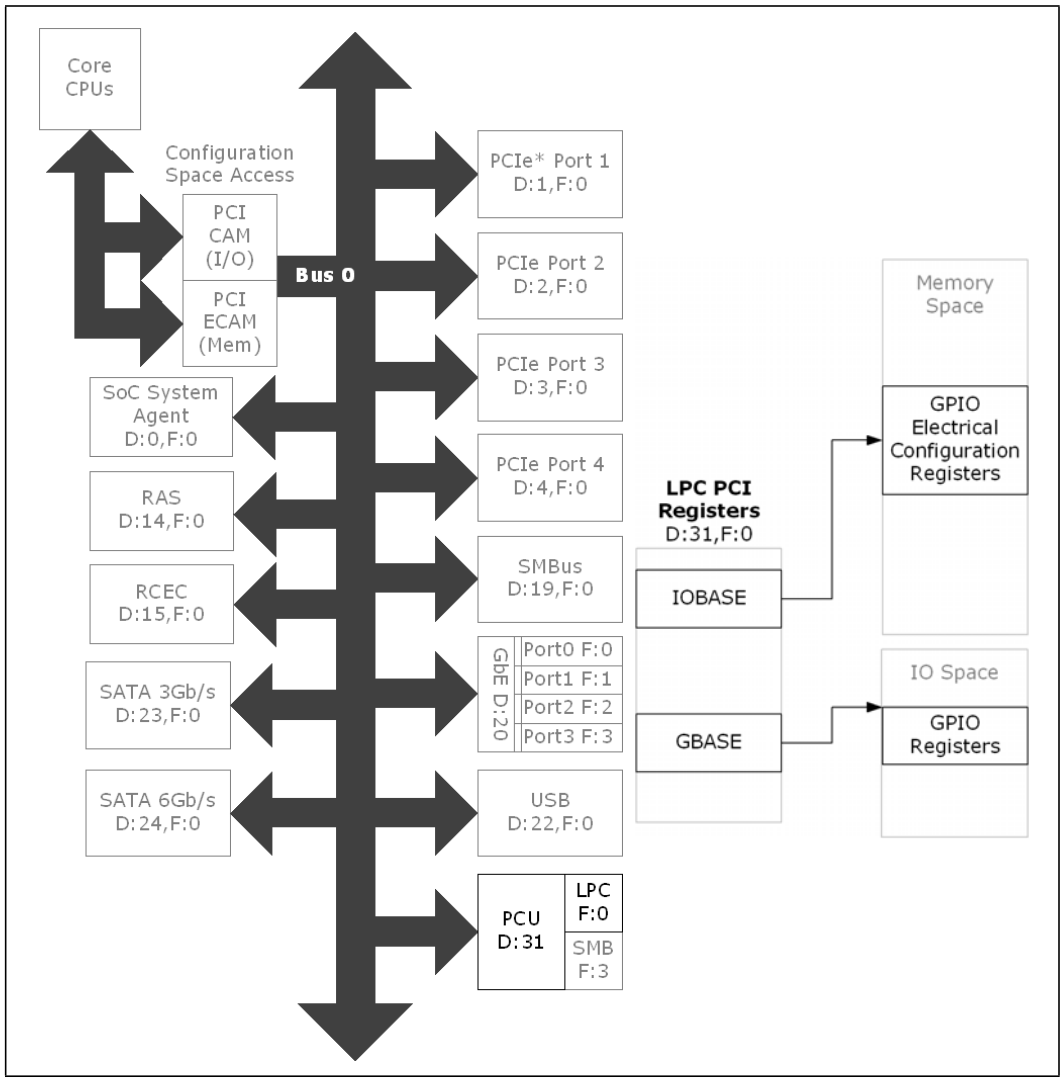
25.3.5 Wake-up Events

GPIO_SUS[3:0] and GPIO_SUS[7:6] also support wake-up events.

25.4 Register Map

Figure 25-2 shows the SoC GPIO registers from a system viewpoint.

Figure 25-2. GPIO Registers



§ §

26 Real Time Clock (RTC)

The SoC contains an MC146818B-compatible Real-Time Clock (RTC) with 256 bytes of battery-backed RAM. The real-time clock performs two key functions—keeping track of the time of day and storing system data even when the system is powered down. The RTC operates on a 32.768-kHz crystal and a 3.3V battery.

The RTC supports two lockable memory ranges. By setting the bits in the configuration space, two 8-byte ranges are locked to read and write accesses. This prevents unauthorized reading of passwords or other system security information.

The RTC supports a date alarm that allows for scheduling a wake-up event up to 30 days in advance.

Figure 26-1. RTC Covered in This Chapter

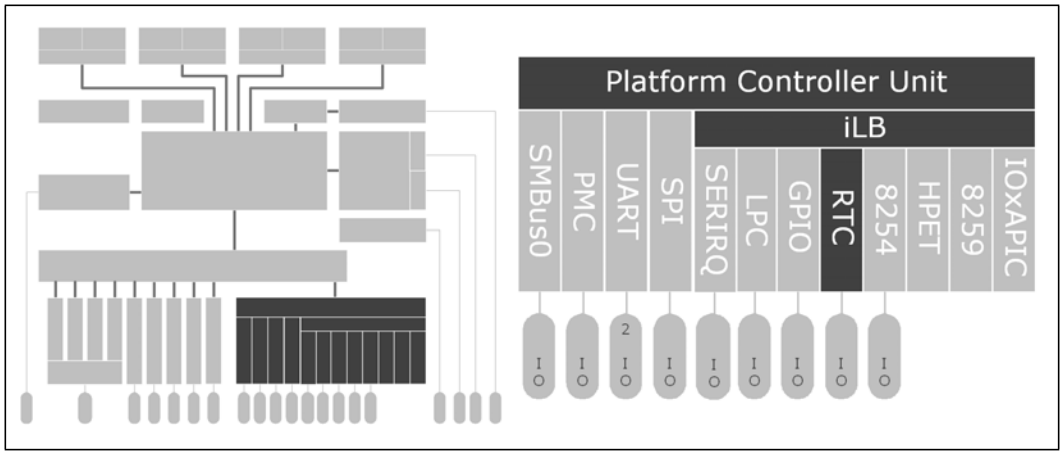


Table 26-1. References

Reference	Revision	Date	Document Title
Intel White Paper 321088	-	January 2009	Accessing the Real Time Clock Registers and the NMI Enable ftp://download.intel.com/design/intarch/PAPERS/321088.pdf

26.1 Signal Descriptions

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

Table 26-2. Signals

Name	Type Direction	Description These signals are in the RTC Power Well
BRTCX1_PAD	I	**Crystal Input 1**: Connects to the 32.768-kHz crystal.
BRTCX2_PAD	O	**Crystal Input 2**: Connects to the 32.768-kHz crystal.
BVCCRTC_EXTPAD	O	Connects to an external 0.1-µF capacitor on the platform board to Vss.
SRTCRST_B	I	**RTC Reset:** When asserted, this active-low input signal resets the SoC register bits in the RTC power well. This signal is normally held high (to VccRTC) through an external RC circuit. The external RC circuit creates a time delay such that SRTCRST_B is deasserted some time after the RTC power is valid to cleanly reset the PMC registers. The RC time delay is in the 10-20 ms range. Cleanly clearing the PMC registers allows the Intel® Trusted Platform Module firmware to detect when the PMC registers have been reset due to battery removal/re-installation. This signal is in the RTC power well. ***Note:*** Unless CMOS is being cleared (only to be done in the G3 power state), the signal input must always be high when all other RTC power planes are on. ***Note:*** In the case where the RTC battery is dead or missing on the platform, the SRTCRST_B signal must be deasserted before the RSMRST_B signal.
RSMRST_B	I	**Resume Well Reset:** Active-low signal located in the RTC power well that resets the SoC circuits located in the Suspend (SUS) power well. An external RC circuit is required to guarantee the SUS well power is valid before RSMRST_B going high.
RTEST_B	I	**RTC Battery Test:** Active-low signal. An external RC circuit creates a time delay for the signal such that it goes high sometime after the battery voltage is valid. The RC time delay must be in the 10-20 ms range. This allows the SoC to detect when a new battery has been installed. This signal is internally asserted after the suspend power is up if the coin cell battery is weak. When active, this signal also resets some bits in the RTC well that are otherwise not reset by PLTRST_B, RSMRST_B, or SRTCRST_B. ***Note:*** This signal may also be used for debug purposes, as part of an XDP port. Unless entering a test mode, the RTEST_B input must always be high when all other non-RTC power planes are on. This signal is in the RTC power well.
COREPWROK	I	Core Power OK: When this input is asserted by the platform board External Circuitry (EC), it is an indication to the SoC that all of its core power rails have been stable for at least 10 ms. COREPWROK is driven asynchronously. When COREPWROK goes low, the SoC asynchronously asserts the active-low PMU_PLTRST_B platform reset signal.

26.2 Features

The RTC features are:

- Similar to digital watches
- Runs even when system unplugged
- Small lithium battery (coin cell) keeps it powered
  - — Typical 6-µA current draw during the system G3 mechanical off state
- 32.768-kHz crystal oscillator
- Seconds, minutes, hours, day, month, year
- Leap-years
- Generates a wake/interrupt when time matches a programmed value
- Includes 242-byte RAM backed by battery (CMOS RAM)
- Registers mapped to fixed I/O locations
- 70h index, 71h data
- Interrupt mapped to IRQ8 in 8259, I/O APIC

26.3 Architectural Overview

The RTC module provides a battery backed-up date and time keeping device. Three interrupt features available are time of day alarm with once-a-second to once-a-month range, periodic rates of 122 ms to 500 ms, and end-of-update cycle notification. Seconds, minutes, hours, days, day of week, month, and year are counted. The hour is represented in a twelve or twenty-four hour format, and data are represented in BCD or binary format. The design is meant to be functionally compatible with the Motorola* MS146818B. The time keeping comes from a 32.768-kHz oscillating source, which is divided to achieve an update every second. The lower 14 bytes on the lower RAM block have specific functions. The first ten are for time and date information. The next four (0Ah to 0Dh) are registers, which configure and report RTC functions. A host-initiated write takes precedence over a hardware update if a collision.

26.3.1 Update Cycles

An update cycle occurs once a second, if the B.SET bit is not asserted and the divide chain is properly configured. During this procedure, the stored time and date are incremented, overflow checked, a matching alarm condition is checked, and the time and date are rewritten to the RAM locations. The update cycle starts at least 488 ms after A.UIP is asserted, and the entire cycle does not take more than 1984 ms to complete. The time and date RAM locations (00h to 09h) are disconnected from the external bus during this time.

26.3.2 Interrupts

The real-time clock interrupt is internally routed within the SoC both to the I/O APIC and the 8259. It is mapped to interrupt vector 8. This interrupt does not leave the SoC, nor is it shared with any other interrupt. IRQ8# from the Serial IRQ (SERIRQ) stream is ignored. However, the High Performance Event Timers (HPET) are also mapped to IRQ8#; in this case, the RTC interrupt is blocked.

26.3.3 Lockable RAM Ranges

The RTC battery-backed RAM supports two locked 8-byte ranges: the Upper 128-Byte Lock (UL) bit in the RTC Configuration (RC) register and the Lower 128-Byte Lock (LL) bit of the RC. When the locking bits are set, the corresponding range in the RAM is not readable or writable. A write cycle to those locations has no effect. A read cycle to those locations does not return the location actual value (resultant value is undefined).

Once a range is locked, the range is unlocked only by a hard reset, which invokes the BIOS and allow it to re-lock the RAM range.

26.4 RTC During Power-Up

The RTC circuitry is always powered-on to update the real time clock. It receives its supply voltage from the connected coin cell. When the power-up sequence begins and the other platform-board power supplies are available, the RTC unit moves to an alternative power supply.

26.5 Clearing the Battery-Backed RTC RAM

Clearing the CMOS RAM in an SoC-based platform is done by using a jumper on SRTCRST_B or through a jumper tied to a dedicated general-purpose I/O port. Do not attempt to clear CMOS by using a jumper to tie Vss to the RTC 3.3V power pin (VCCRTC_3P3).

26.5.1 Using SRTCRST_B to Clear CMOS Registers

A jumper on SRTCRST_B clears the CMOS values as well as resets to default, the state of those configuration bits that reside in the RTC power well. When the SRTCRST_B is strapped to ground, the bit of the RTC_PWR_STS (RPS) bit of the General PM Configuration 1 (GEN_PMCON1) register is set and those configuration bits in the RTC power well are set to their default state. The BIOS monitors the state of this bit and manually clears the RTC CMOS array once the system is booted. The normal position causes SRTCRST_B to be pulled up through a weak pull-up resistor. Table 26-3 shows which bits are set to their default state when SRTCRST_B is asserted.

This SRTCRST_B jumper technique allows the jumper to be moved and then replaced while the system is powered-off. Then, once booted, the GEN_PMCON1.RPS bit is detected in the set state.

Table 26-3. Register Bits Reset by Asserting SRTCRST_B

Register Bit	Bit(s)	Default State
RCRB_GENERAL_CONTROL.TS	1	xb
GEN_PMCON1.PME_B0_S5_DIS	15	0b
GEN_PMCON1.WOL_EN_OVRD	13	0b
GEN_PMCON1.DIS_SLP_X_STRCH_SUS_UP	12	0b
GEN_PMCON1.RTC Reserved	8	0b
GEN_PMCON1.SWSMI_RATESEL	7:6	00b
GEN_PMCON1.S4MAW	5:4	00b
GEN_PMCON1.S4ASE	3	0b
GEN_PMCON1.AG3E	0	1b
PM1_STS_EN.RTC_EN	26	0b
PM1_STS_EN.PWRBTNOR_STS	11	0b
PM1_CNT.SLP_TYP	12:10	0b
GPE0a_EN.PME_B0_EN	13	0b
GPE0a_EN.BATLOW_EN	10	0b

26.5.2 Using a GPI to Clear CMOS Registers

A jumper on a General-Purpose Input (GPI) pin is also used to clear the CMOS values. The BIOS detects the setting of this GPI on system boot-up and manually clears the CMOS array.

Note: The GPI strap technique to clear the CMOS requires multiple steps to implement. The system is booted with the jumper in the new position, then powered back down. The jumper is replaced back to the normal position, then the system is rebooted again.

Warning: Do not implement a jumper on VCCRTC_3P3 to clear the CMOS.

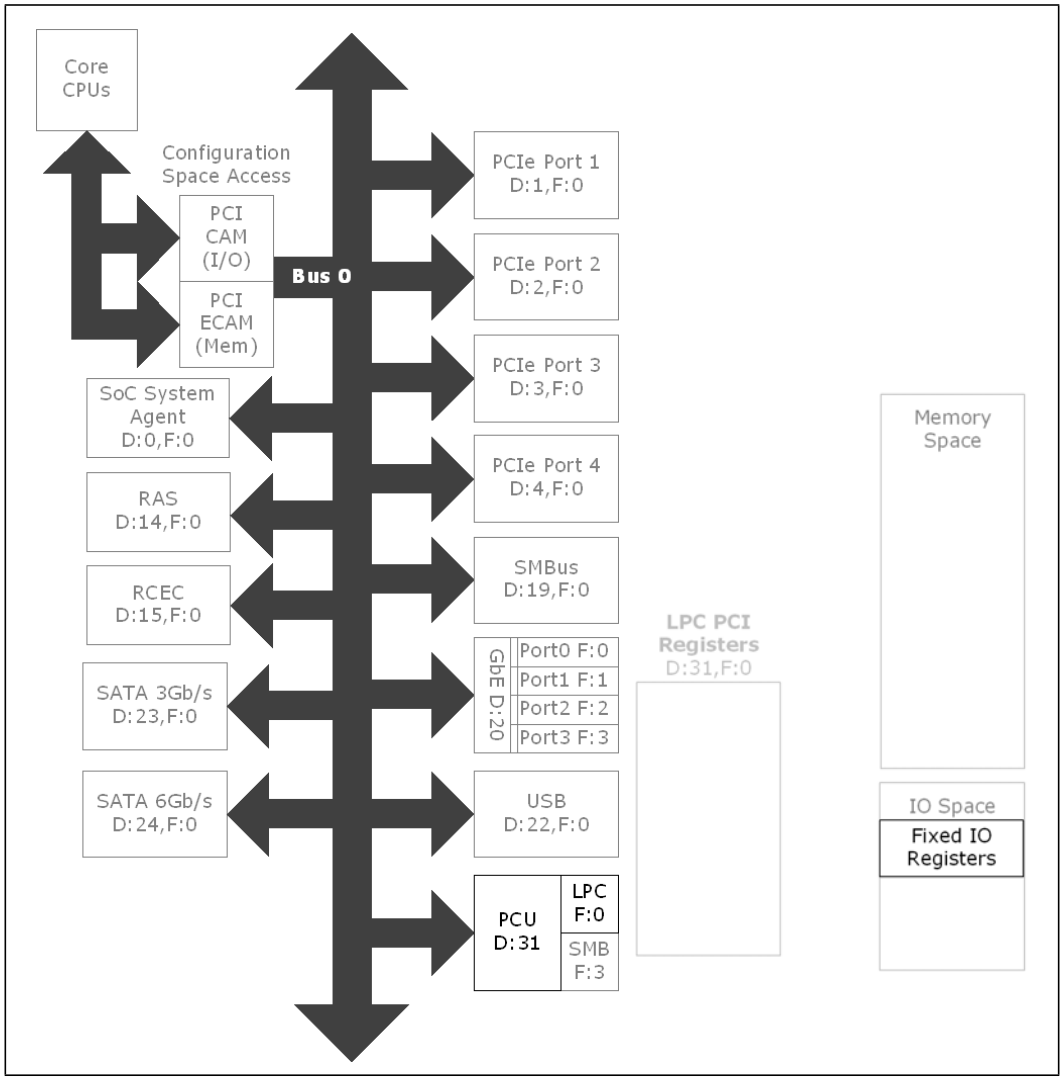
26.6 Support of S0idle Power-Saving Mechanism

The RTC interface is active during the S0idle state.

26.7 Register Map

Figure 26-2 shows the SoC RTC registers from a system viewpoint.

Figure 26-2. RTC Register Map



26.7.1 Registers in I/O Space

The RTC internal registers and RAM are organized as two banks of 128 bytes each, called the standard and extended banks.

Note: Disabling the extended bank does not occur.

The first 14 bytes of the standard bank contain the RTC time and date information along with four registers, A-D, that are used for configuration of the RTC. The extended bank contains a full 128 bytes of battery backed SRAM. All data movement between the host CPU and the RTC is done through registers mapped to the standard I/O space.

The Indexed Registers (IR) and Target Registers (TR) are used for data movement to and from the standard bank. The Extended RAM Index Register (RIR) and Extended RAM Target Register (RTR) are used for data movement to and from the extended bank. All of these registers have alias I/O locations as indicated in Table 26-4.

Table 26-4. RTC Registers in I/O Space

I/O Address	Alias I/O Location	If U128E Bit = 0	Default Value	Name	Description
0x70	74h	Also alias to 72h and 76h	00h	IR	Indexed Registers
0x71	75h	Also alias to 73h and 77h	00h	TR	Target Registers
0x72	76h		00h	RIR	Extended RAM Index Register (if enabled)
0x73	77h		00h	RTR	Extended RAM Target Register (if enabled)

Note: Writes to 72h, 74h, and 76h do not affect the NMI enable (bit 7 of 70h).

I/O locations 70h and 71h are standard ISA locations for the real time clock. Locations 72h and 73h are for accessing extended RAM. The extended RAM bank is also accessed using an indexed scheme. The I/O address 72h is used as the address pointer, and I/O address 73h is used as the data register. Index addresses above 127h are not valid.

26.7.2 Difficulty Accessing These Registers

The registers needed to access the two banks of RTC registers at the I/O locations 0x70-0x77 have been present since the early days of personal computers. They were originally located in the real time clock circuitry itself before being absorbed into Intel silicon as PC designs matured. As such, it retains a lot of the legacy limitations inherent with earlier architectures (aliasing, etc.).

Accessing these registers is difficult. The NMI Enable bit (NMI_EN = I/O 0x70[7]) is especially troublesome as a straight read of this register returns all 0xFF data, although writes work fine. Refer to the Intel white paper for guidance:

Accessing the Real Time Clock Registers and the NMI Enable
ftp://download.intel.com/design/intarch/PAPERS/321088.pdf

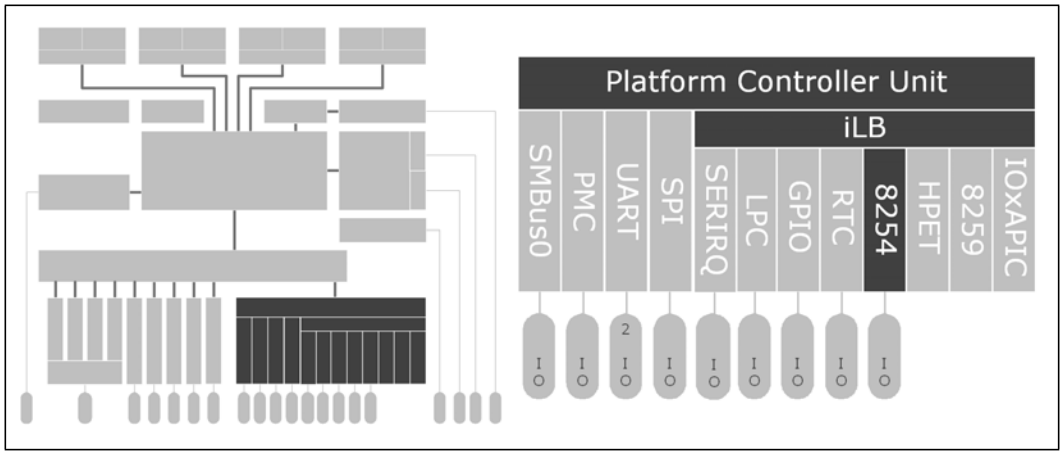
This paper explains the details and the necessary steps the software needs to perform to access these registers.

§ §

27 8254 Programmable Interval Timer (PIT)

The SoC 8254 Programmable Interval Timer (PIT) contains three counters which have fixed legacy uses including the system timer, the DRAM refresh timer, and the speaker tone. The DRAM refresh timer is not used by the SoC.

Figure 27-1. 8254 PIT Covered in This Chapter



27.1 Signal Descriptions

Besides the 14.31818-MHz clock supplied to the SoC, one signal pin is associated with the integrated 8254 PIT.

The signal description table has the following headings:

- **Signal Name:** The name of the signal/pin
- **Direction:** The buffer direction is either input, output, or I/O (bi-directional)
- **Type:** The buffer type
- **Description:** A brief explanation of the signal function

Table 27-1. Signals

Signal Name	Direction/ Type	Description
SPKR	OD	Speaker: The signal drives an external speaker driver device, which in turn drives the system speaker. Upon PMU_PLTRST_B, its output state is 0. *This signal is muxed with GPIOS_12 and SMB_CLK1.*

27.2 Features

Accessed through 8-bit registers mapped to the I/O space using 16 address bits:

- I/O addresses 0x40-0x43, aliased to 0x50-0x53

Has three internal counters/timers:

- 8254 Timer 0 - Used for OS timer tick.
  - — Mapped to IRQ0 on 8259, IRQ2 on I/O APIC.
  - — Typically set to 1-ms to 50-ms period.
- 8254 Timer 1 - No longer used (was for ISA refresh).
- 8254 Timer 2 - “Beep” speaker.

27.3 Architectural Overview

The 8254 Programmable Interval Timer (PIT) contains three counters which have fixed uses. All registers and functions associated with the 8254 timers are in the core power well and are clocked by the SoC 14.31818-MHz clock.

This clock is divided by 12 internally to generate the 1.193182-MHz (838-ns period) reference clock used by the three counters.

The PIT is accessed through a set of four registers located in the I/O space.

- One 8-bit 8254 Timer Control Word Register
- Three 8-bit Counter Access Ports, one for each counter

They are accessed in the I/O space and have been assigned the 16-bit I/O addresses of 0x40 through 0x43.

Also associated with the integrated 8254 are 4 bits of the 8-bit NMI Status and Control (NSC) register at address 0x61 in the I/O space.

27.3.1 Timer Control Word (TCW)

I/O address 0x43h: TCW - Timer Control Word Register

The TCW is programmed before any counter being accessed to specify the counter modes. Following reset, the control words for each counter are undefined and each counter output is 0. Each timer must be programmed to bring it into a known state.

Two special commands are issued to the three counters through the TCW register. When these commands are chosen, several bits within the TCW register are redefined.

- Read Back command
- Counter Latch command

27.3.1.1 Read Back Command

This command determines the count value, programmed mode, and current states of the OUT pin and null count flag of the selected counter or counters. Status and/or count are latched in any or all of the counters by selecting the counter during the register write. The count and status remain latched until read, and further latch commands are ignored until the count is read.

Both count and status of the selected counters are latched simultaneously by setting both bit 5 and bit 4 to 0. If both are latched, the first read operation from that counter returns the latched status. The next one or two reads, depending on whether the counter is programmed for 1- or 2-byte counts, returns the latched count.

See Section 27.4, “Programming the 8254 Counters” on page 533 for additional programming information.

27.3.1.2 Counter Latch Command

This command latches the current count value and ensures the count read from the counter is accurate. The count value is then read from each counter count register through the Counter Ports Access Ports Register (40h for counter 0, 41h for counter 1, and 42h for counter 2). The count must be read according to the programmed format, i.e., if the counter is programmed for 2-byte counts, 2 bytes must be read.

The 2 bytes do not have to be read one right after the other (read, write, or programming operations for other counters are inserted between the reads). If a counter is latched once and then latched again before the count is read, the second Counter Latch command is ignored.

See Section 27.4, "Programming the 8254 Counters" on page 533 for additional programming information.

27.3.2 Counter 0, System Timer

I/O address 0x40h: Counter 0

This counter functions as the system timer by controlling the state of IRQ0 and is programmed for mode 3 operation.

The counter produces a square wave with a period equal to the product of the counter period (838 ns) and the initial count value. The counter loads the initial count value one counter period after the software writes the count value to the counter I/O address. The counter initially asserts IRQ0 and decrements the count value by two each counter period. The counter negates IRQ0 when the count value reaches 0. It then reloads the initial count value and again decrements the initial count value by two each counter period. The counter then asserts IRQ0 when the count value reaches 0, reloads the initial count value, and repeats the cycle, alternately asserting and negating IRQ0.

27.3.3 Counter 1, Refresh Request Signal

I/O address 0x41h: Counter 1

This counter is programmed for mode 2 operation and impacts the period of the Refresh Cycle Toggle Status (RTS) bit of the NMI Status and Control (NSC) register at address 0x61 in the I/O space. Programming the counter to anything other than mode 2 results in undefined behavior. See Table 27-2 for the NSC bit definitions.

27.3.4 Counter 2, Speaker Tone

I/O address 0x42h: Counter 2

This counter provides the speaker tone and is typically programmed for mode 3 operation. The counter provides a speaker frequency equal to the counter clock frequency (1.193 MHz) divided by the initial count value. The speaker must be enabled by a write to the NSC.SDE register bit. See Table 27-2 for the NSC bit definitions.

27.3.5 NMI Status and Control (NSC)

I/O address 0x61h: NSC

Four bits of the NSC register are associated with the integrated 8254 PIT.

Table 27-2. NSC Register Bits Used by the 8254 PIT

NSC Bit	Name	Long Name	Description
5	T2S	Timer Counter 2 Status	Reflects the current state of the 8254 Counter 2 outputs.
4	RTS	Refresh Cycle Toggle Status	Reflects the current state of 8254 Counter 1.
1	SDE	Speaker Data Enable	When this bit is a 0, the SPKR output is a 0. When this bit is a 1, the SPKR output is equivalent to the Counter 2 OUT signal value.
0	TC2E	Timer Counter 2 Enable	When cleared, Counter 2 counting is disabled. When set, counting is enabled.

27.4 Programming the 8254 Counters

The counter/timers are programmed in the following fashion:

1. Write a control word to select a counter.
2. Write an initial count for that counter.
3. Load the least and/or most significant bytes (as required by control word bits 5, 4) of the 16-bit counter.
4. Repeat with the other counters.

Only two conventions need to be observed when programming the counters. First, for each counter, the control word must be written before the initial count is written. Second, the initial count must follow the count format specified in the control word (least significant byte only, most significant byte only, or least significant byte and then most significant byte).

A new initial count is written to a counter at any time without affecting the counter programmed mode. Counting is affected as described in the mode definitions. The new count must follow the programmed count format.

If a counter is programmed to read/write 2-byte counts, the precaution applies as follows: a program must not transfer control between writing the first and second byte to another routine which also writes into that same counter. Otherwise, the counter is loaded with an incorrect count.

The Control Word Register at port 43h controls the operation of all three counters. Several commands are available as follows:

- Control Word command - Specifies which counter to read or write, the operating mode, and the count format (binary or binary-coded decimal).
- Counter Latch command - Latches the current count so the system reads it. The countdown process continues.
- Read Back command - Reads the count value, programmed mode, the current state of the OUT pins, and the state of the null count flag of the selected counter.

Table 27-3 lists the six operating modes for the interval counters.

Table 27-3. Counter Operating Modes

Mode	Function	Description
0	Out signal on end of count (=0)	Output is 0. When the count goes to 0, the output goes to 1 and stays at 1 until the counter is reprogrammed.
1	Hardware-retriggerable one-shot	Output is 0. When the count goes to 0, the output goes to 1 for one clock time.
2	Rate generator (divide by n counter)	Output is 1. The output goes to 0 for one clock time, then back to 1 and the counter is reloaded.
3	Square wave output	Output is 1. The output goes to 0 when the counter rolls over, and the counter is reloaded. The output goes to 1 when the counter rolls over, and the counter is reloaded, etc.
4	Software triggered strobe	Output is 1. The output goes to 0 when the count expires for one clock time.
5	Hardware triggered strobe	Output is 1. The output goes to 0 when the count expires for one clock time.

27.5 Reading from the Interval Timer

It is often desirable to read the value of a counter without disturbing the count in progress. The three methods for reading the counters are as follows: a simple read operation, the Counter Latch command, and the Read-back command. They are explained below.

With the simple read and Counter Latch command methods, the count must be read according to the programmed format; specifically, if the counter is programmed for 2-byte counts, 2 bytes must be read. The 2 bytes do not have to be read one right after the other. Read, write, or programming operations for other counters are inserted between them.

27.5.1 Simple Read

The first method is to perform a simple read operation. The counter is selected through port 40h (Counter 0), 41h (Counter 1), or 42h (Counter 2).

Note: Performing a direct read from the counter does not return a determinate value, because the counting process is asynchronous to read operations. However, in the case of Counter 2, the count is stopped by writing 0b to the NSC.TC2E register bit. See Table 27-2 on page 532 for the NSC bit definitions.

27.5.2 Counter Latch Command

The Counter Latch command, written to port 43h, latches the count of a specific counter at the time the command is received. This command ensures that the count read from the counter is accurate, particularly when reading a 2-byte count. The count value is then read from each counter Count register that was programmed by the Control register.

The count is held in the latch until it is read or the counter is reprogrammed. The count is then unlatched and allows reading the contents of the counters quickly without affecting the counting in progress. Multiple counter latch commands are used to latch more than one counter. Counter latch commands do not affect the programmed mode of the counter in any way.

When a counter is latched and later latched again before the count is read, the second Counter Latch command is ignored. The first Counter Latch command issued is the first count read.

27.5.3 Read-Back Command

The Read-back command, written to port 43h, latches the count value, programmed mode, and current states of the OUT pin and null count flag of the selected counter or counters. The value of the counter and its status may then be read by I/O access to the counter address.

The Read-back command latches multiple counter outputs at one time. This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter latched count is held until it is read or reprogrammed. Once read, a counter is unlatched. The other counters remain latched until they are read. If multiple count read-back commands are issued to the same counter without reading the count, all but the first are ignored.

The Read-back command may additionally be used to latch status information of selected counters. The status of a counter is accessed by a read from that counter I/O port address. If multiple counter status latch operations are performed without reading the status, all but the first are ignored.

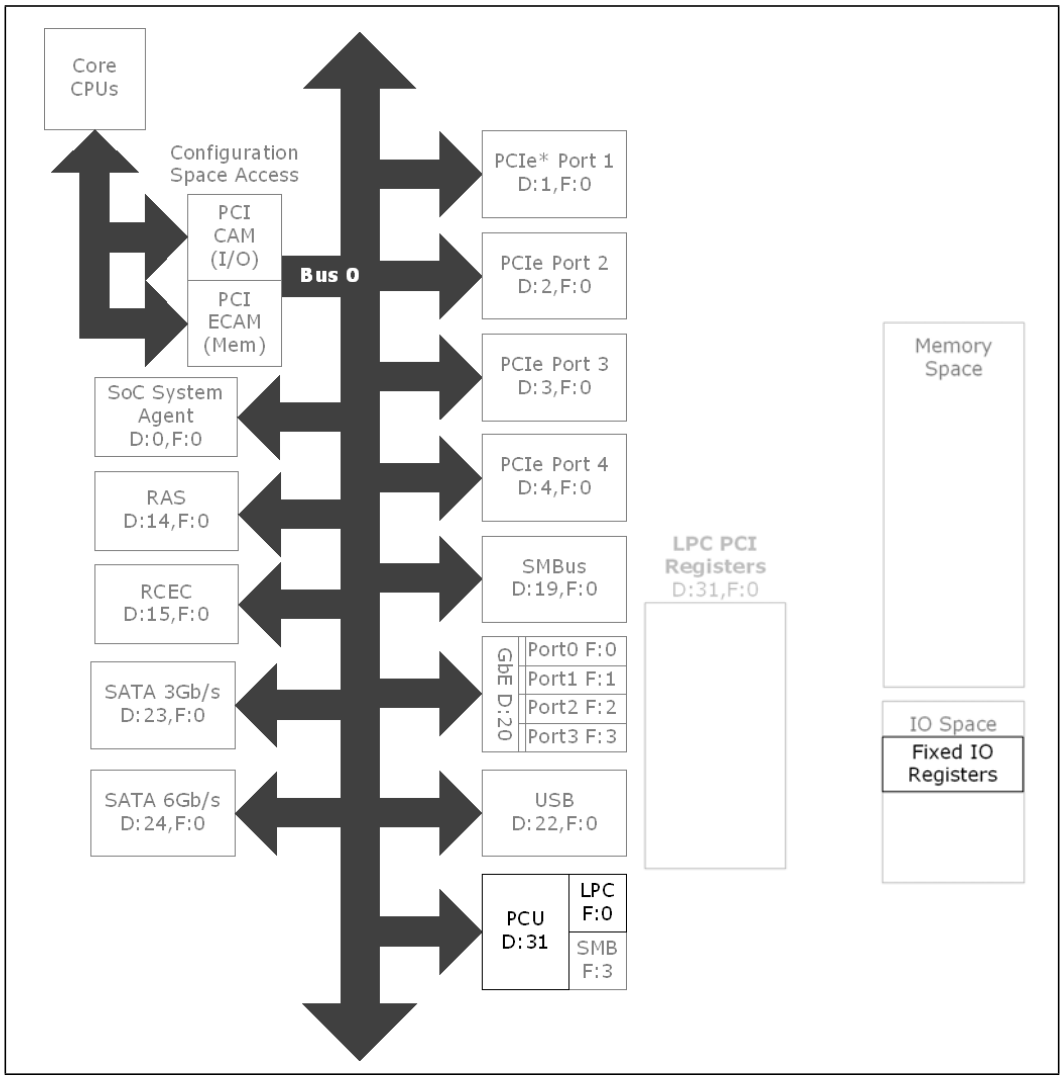
Both count and status of the selected counters are latched simultaneously. This is functionally the same as issuing two consecutive, separate read-back commands. If multiple count and/or status read-back commands are issued to the same counters without any intervening reads, all but the first are ignored.

If both count and status of a counter are latched, the first read operation from that counter returns the latched status, regardless of which was latched first. The next one or two reads, depending on whether the counter is programmed for one or two type counts, returns the latched count. Subsequent reads return the unlatched count.

27.6 Register Map

Figure 27-2 shows the SoC 8254 PIT registers from a system viewpoint.

Figure 27-2. 8254 PIT Register Map



27.6.1 I/O Mapped Registers

The I/O ports listed in Table 27-4 have multiple register functions depending on the current programmed state of the 8254. The port numbers referenced in the register descriptions following Table 27-4 is one combination but not the only one.

Table 27-4. Register Aliases

Port	Alias	Register Name	Default Value	Access
40h	50h	Counter 0 Interval Time Status Byte Format (C0TS)	0xxxxxxxb	RO
		Counter 0 Counter Access Port Register (C0AP)	Undefined	RW
41h	51h	Counter 1 Interval Time Status Byte Format (C1TS)	0xxxxxxxb	RO
		Counter 1 Counter Access Port Register (C1AP)	Undefined	RW
42h	52h	Counter 2 Interval Time Status Byte Format (C2TS)	0xxxxxxxb	RO
		Counter 2 Counter Access Port Register (C2AP)	Undefined	RW
43h	53h	Timer Control Word Register (TCW)	Undefined	WO
		Read Back Command (RBC)	xxxxxxx0b	WO
		Counter Latch Command (CLC)	xxxx0000b	WO

Table 27-5. 8254 PIT Registers in I/O Space

Address in I/O Space	Name	Long Name
0x40	C0TS	Counter 0 Interval Time Status Byte Format
0x41	C1TS	Counter 1 Interval Time Status Byte Format
0x42	C2TS	Counter 2 Interval Time Status Byte Format
0x43	TCW	Timer Control Word Register • Read Back Command (RBC) • Counter Latch Command (CLC)
0x50	C0AP	Counter 0 Counter Access Port Register
0x51	C1AP	Counter 1 Counter Access Port Register
0x52	C2AP	Counter 2 Counter Access Port Register
0x61	NSC	NMI Status and Control

§ §

28 High Precision Event Timer (HPET)

The High Precision Event Timer (HPET) provides a set of timers that are used by the operating system for timing events. One timer block is implemented, containing one counter and three timers.

It complies with the *IA-PC HPET (High Precision Event Timers) Specification*, Revision 1.0.

Figure 28-1. HPET Covered in This Chapter

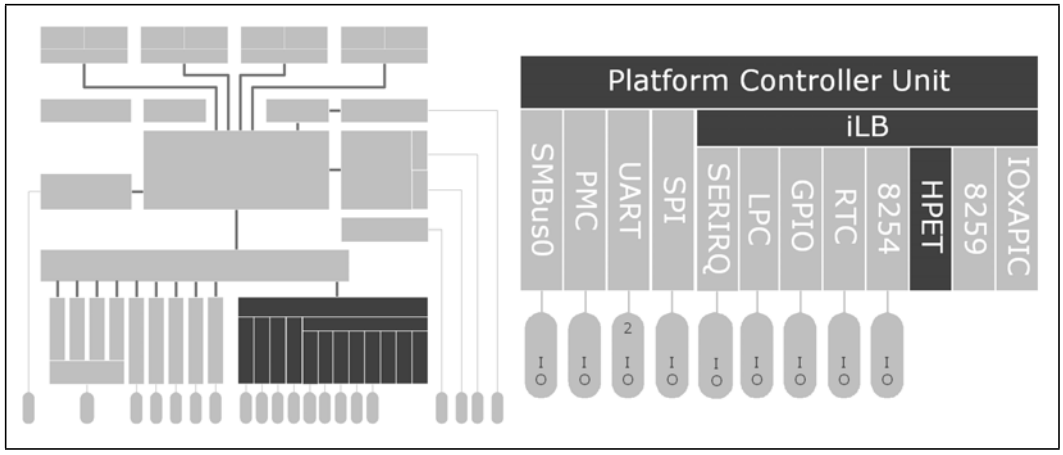


Table 28-1. References

Reference	Revision	Date	Document Title
HPET	1.0a	October 2004	*IA-PC HPET (High Precision Event Timers) Specification*, Revision 1.0a

The *IA-PC HPET Specification* is available at: http://www.intel.com/content/www/us/en/software-developers/software-developers-hpet-spec-1-0a.html.

28.1 Signal Descriptions

No signal pins/balls are associated with the HPET.

28.2 Features

The HPET features are as follows:

- Includes one periodic timer, 2 one-shots (total three comparators).
- Improves resolution, reduces overhead.
- Results in fewer interrupts to the CPU.

28.3 Architectural Overview

This function provides a set of timers that are used by the operating system.

Three timers are implemented as a single counter each with its own comparator and value register. This counter increases monotonically. Each individual timer generates an interrupt when the value in its value register matches the value in the main counter.

The registers associated with these timers are mapped to a memory space at fixed, 32-bit addresses of 0xFED00000 through 0xFED003FF. Some portions of this address range are not used and are reserved.

28.3.1 Configuration Registers

Each timer is individually configured through memory addresses show in Table 28-2.

Table 28-2. Timer Configuration in Memory Space

Address in Memory Space	Default Value	Name	Description
0xFED00100	00F0_0000_0000_0030h	HPET_T0C	Timer 0 Configuration and Capabilities
0xFED00120	00F0_0000_0000_0000h	HPET_T1C	Timer 1 Configuration and Capabilities
0xFED00140	00F0_0800_0000_0000h	HPET_T2C	Timer 2 Configuration and Capabilities

28.3.2 Timer Comparator

Memory reads to the registers show in Table 28-3 return the current value of the comparator. The default value for each timer is all 1s for the bits that are implemented.

- Timer 0 is 64-bits wide.
- Timers 1 and 2 are 32-bits wide.

Table 28-3. Timer Comparator Values

Address in Memory Space	Default Value	Name	Description
0xFED00108	FFFF_FFFFh	HPET_T0CV_L	Lower Timer 0 Comparator Value
0xFED0010C	FFFF_FFFFh	HPET_T0CV_U	Upper Timer 0 Comparator Value
0xFED00128	0000_0000_FFFF_FFFFh	HPET_T1CV	Timer 1 Comparator Value
0xFED00148	0000_0000_FFFF_FFFFh	HPET_T2CV	Timer 2 Comparator Value

28.3.3 Interrupts

The General Interrupt Status (HPET_GIS) register provides 1 bit for each of the three counters. In edge-triggered mode, this bit always reads as 0. In level-triggered mode, this bit is set when an interrupt is active.

28.3.4 Timer Accuracy

The General Capabilities and ID (HPET_GCID) register indicates that the HPET Counter Tick Period (CTP) is 69.841279 ns (the period of a 14.1318-MHz clock).

1. The timers are accurate over any 1-ms period to within 0.05% of the time specified in the timer resolution fields.
2. Within any 100-ms period, the timer reports a time that is up to two ticks too early or too late. Each tick is less than or equal to 100 ns, so this represents an error of less than 0.2%.
3. The timer is monotonic. It does not return the same value on two consecutive reads (unless the counter has rolled over and reached the same value).

The main counter is clocked by the 14.31818-MHz clock. The accuracy of the main counter is as accurate as the 14.31818-MHz clock.

28.4 Programming the HPET

28.4.1 Non-Periodic Mode - All Timers

Each timer supports the non-periodic mode of operation. This mode is thought of as creating a one-shot. When a timer is set up for the non-periodic mode, it generates an interrupt when the value in the main counter matches the value in the timer comparator register. As timers 1 and 2 are 32-bit, they generate another interrupt when the main counter wraps.

The 64-bit Timer 0 Comparator Value (T0CV) cannot be programmed reliably by a single 64-bit write in a 32-bit environment unless only the periodic rate is being changed. If T0CV needs to be re-initialized, the algorithm is performed as follows:

1. Set the Timer Value Set (TVS) - T0C.TVS.
2. Set T0CV[31:0].
3. Set the TVS of the Timer 0 Configuration and Capabilities (TOC) - T0C.TVS.
4. Set T0CV[63:32].

28.4.2 Periodic Mode - Timer 0 Only

When set up for the periodic mode when the main counter value matches the value in the Timer 0 Comparator Value (T0CV), an interrupt is generated (if enabled). The hardware then increases T0CV by the last value written to T0CV. During run-time, T0CV is read to find out when the next periodic interrupt is generated. The software is expected to remember the last value written to T0CV.

Example: if the value written to T0CV is 00000123h, then:

- An interrupt is generated when the main counter reaches 00000123h.
- T0CV is then adjusted to 00000246h.
- Another interrupt is generated when the main counter reaches 00000246h.
- T0CV is then adjusted to 00000369h.

When the incremented value is greater than the maximum value for T0CV, the value wraps around through 0. For example, if the current value in a 32-bit timer is FFFF0000h and the last value written to this register is 20000, then after the next interrupt the value changes to 00010000h.

If the software wants to change the periodic rate, it writes a new value to T0CV. When the timer comparator matches, the new value is added to derive the next matching point. If the software resets the main counter, the value in the Comparator Value register must also be reset by setting T0C.TVS. To avoid race conditions, this is done with the main counter halted. The usage model is expected as follows:

1. The software clears the Overall Enable (EN) bit of the General Configuration (HPET_GCFG) register to prevent any interrupts.
2. The software clears the main counter by writing a value of 00h to it.
3. The software sets T0C.TVS.
4. The software writes the new value in T0CV.
5. The software sets HPET_GCFG.EN to enable interrupts.

28.4.3 Programming Timer Interrupts

If each timer has a unique interrupt and the timer has been configured for edge-triggered mode, then no specific steps are required. If configured to level-triggered mode, then its interrupt must be cleared by the software by writing a 1 back to the bit position for the interrupt to be cleared.

Interrupts associated with the various timers have several interrupt mapping options. The software masks GCFG.LRE when reprogramming the HPET interrupt routing to avoid spurious interrupts.

28.4.3.1 Mapping Option #1: Legacy Option (GCFG.LRE Set)

When set, Legacy Rout Enable (LRE) of the General Configuration (HPET_GCFG) register forces the following mapping shown in Table 28-4.

Table 28-4. Legacy Routing

Timer	8259 Mapping	I/O APIC Mapping	Comment
0	IRQ0	IRQ2	The 8254 PIT interrupt is blocked and does not cause any interrupts.
1	IRQ8	IRQ8	The RTC interrupt is blocked and does not cause any interrupts.
2	IRQ11	T2C.IRC and T2C.IR	For I/O APIC mapping, the interrupt is mapped to one of the interrupts indicated by the value of Interrupt Rout Capability (IRC) of the TC2 register. This is hardwired to indicate support for I/O APIC interrupts IRQ11, 20, 21, 22, and 23. The 5-bit value of T2C.IR indicates which of the capable interrupts are used.

When LRE is set, T0C.IR and T1C.IR have no impact for timers 0 and 1.

28.4.3.2 Mapping Option #2: Standard Option (GCFG.LRE Cleared)

When cleared, Legacy Rout Enable (LRE) of the General Configuration (HPET_GCFG) register forces each timer to indicate its own routing control. The interrupts are routed to various interrupts in the I/O APIC. The Interrupt Rout Capability (IRC) bit of each Timer Configuration (TCx) register indicates which interrupts are valid options for routing. The 5-bit value of each counter T2C.IR field indicates which of the capable interrupts are used.

If a timer is set for the edge-triggered mode, the timers are not shared with any other interrupts.

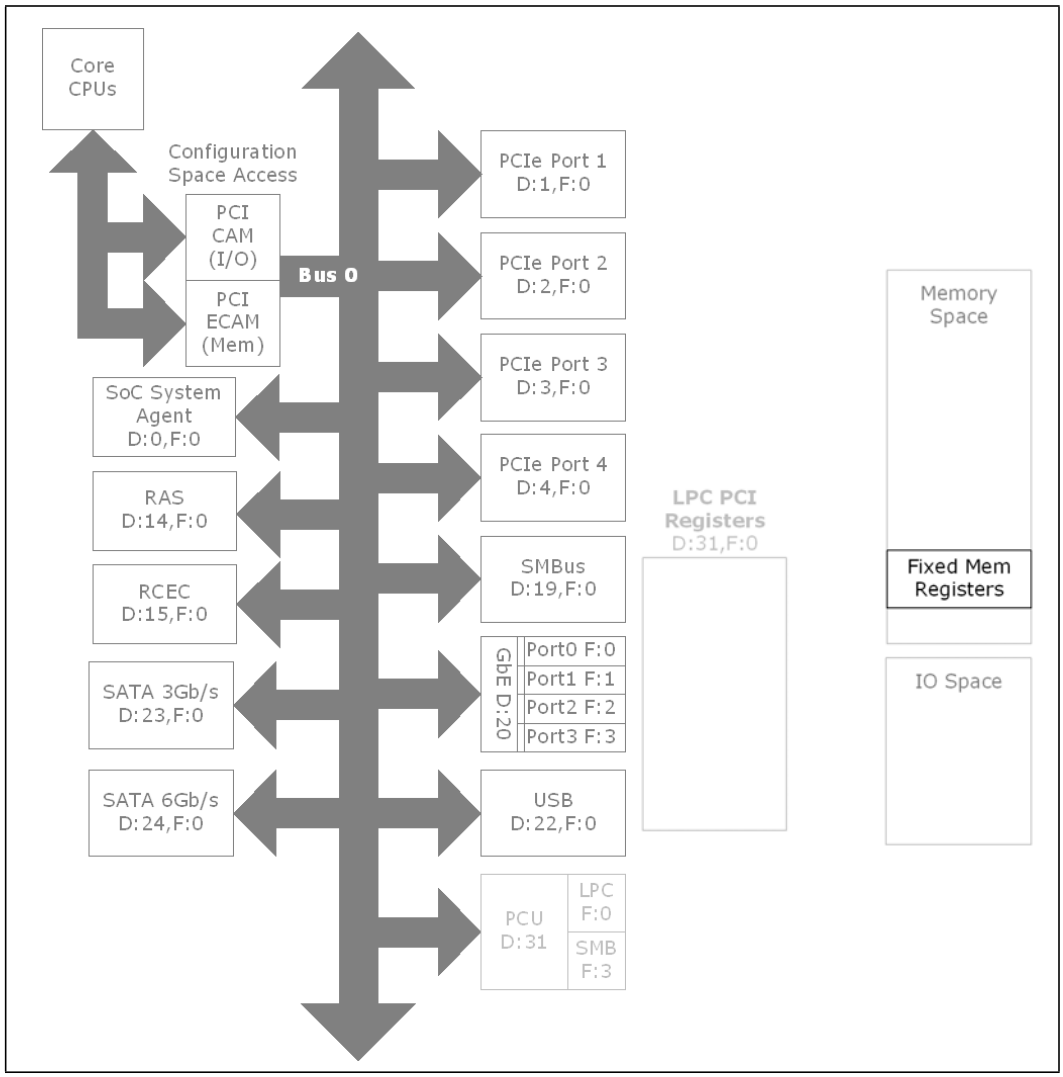
28.4.4 Support of S0idle Power-Saving Mechanism

The HPET is active and keeps running during the S0idle state.

28.5 Register Map

Figure 28-2 shows the SoC HPET registers from a system viewpoint.

Figure 28-2. HPET Register Map



28.5.1 Memory-Mapped Registers

The register space is memory mapped to a 1-KB block starting at address FED0_0000h. All registers are in the core power well. Accesses that cross the register boundaries result in undefined behavior.

Table 28-5. HPET Registers in Memory Space

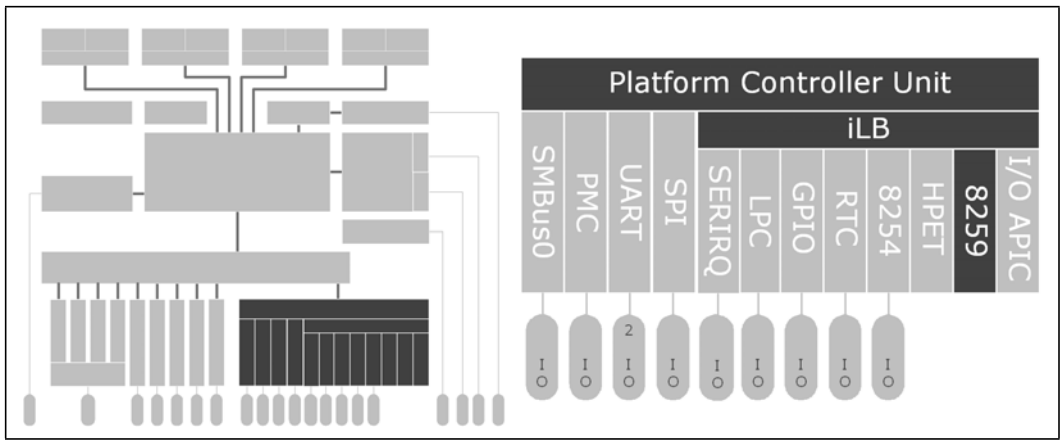
Address in Memory Space	Name	Description
0xFED00000	HPET_GCID	General Capabilities and ID
0xFED00010	HPET_GCFG	General Configuration
0xFED00020	HPET_GIS	General Interrupt Status
0xFED000F0	HPET_MCV	Main Counter Value
0xFED00100	HPET_T0C	Timer 0 Configuration and Capabilities
0xFED00108	HPET_T0CV_L	Lower Timer 0 Comparator Value
0xFED0010C	HPET_T0CV_U	Upper Timer 0 Comparator Value
0xFED00120	HPET_T1C	Timer 1 Configuration and Capabilities
0xFED00128	HPET_T1CV	Timer 1 Comparator Value
0xFED00140	HPET_T2C	Timer 2 Configuration and Capabilities
0xFED00148	HPET_T2CV	Timer 2 Comparator Value

§ §

29 8259 Programmable Interrupt Controller (PIC)

The SoC provides an ISA-Compatible Programmable Interrupt Controller (PIC). It consists of two integrated, cascaded 8259 interrupt controllers.

Figure 29-1. 8259 PIC Covered in This Chapter



29.1 Signal Descriptions

No external signal pins are associated with the integrated 8259 PICs.

29.2 Architectural Overview

In addition to providing support for ISA compatible interrupts, this interrupt controller also supports PCI-based interrupts (PIRQs) by mapping the PCI interrupt onto a compatible ISA interrupt line. Each 8259 PIC supports eight interrupts, numbered 0–7. Table 29-1 shows how the controllers are connected.

Note: The SoC does not provide external PIRQ# signal pins. The PIRQs referred to in this chapter originate from the SoC internal interrupt-routing unit.

How PIRQA, PIRQB, PIRQC, PIRQD, PIRQE, PIRQF, PIRQG, and PIRQH, are routed to IRQ inputs is determined by a set of eight PIRQx Routing Control registers located in the ILB memory-mapped I/O space, offset 0x08-0x0F. This and all SoC interrupt routing are presented in Section 6, “Interrupt Architecture” on page 115.

Table 29-1. 8259 PIC Input Mapping (Sheet 1 of 2)

I/O PIC Input	Master or Slave 8259 PIC Input	Interrupts Routed to This PIC Input	Note
IRQ0	Master IRQ0	• HPET Timer 0 (if GCFG.LRE is set) • 8254 Timer (if GCFG.LRE is not set)	1
IRQ1	Master IRQ1	• SERIRQ (1)	
IRQ2	Master IRQ2	INTR output of the slave 8259 PIC	1, 2
IRQ3	Master IRQ3	• SERIRQ (3), or • UART COM2, or • PIRQx	3
IRQ4	Master IRQ4	• SERIRQ (4), or • UART COM1, or • PIRQx	3
IRQ5	Master IRQ5	• SERIRQ (5), or • GPIO, or • PIRQx	3
IRQ6	Master IRQ6	• SERIRQ (6), or • GPIO, or • PIRQx	3
IRQ7	Master IRQ7	• SERIRQ (7), or • GPIO, or • PIRQx	3
IRQ8	Slave IRQ0	• HPET Timer 1 (if GCFG.LRE is set) • RTC (if GCFG.LRE is not set)	1
IRQ9	Slave IRQ1	• SERIRQ (9), or • PIRQx, or • From SCI (based on the ACTL.SCIS and PM1_CNT.SCI_EN registers)	3
IRQ10	Slave IRQ2	• SERIRQ (10), or • PIRQx, or • From SCI (based on the ACTL.SCIS and PM1_CNT.SCI_EN registers)	3
IRQ11	Slave IRQ3	• SERIRQ (11), or • HPET Timer 2, or • PIRQx, or • From SCI (based on the ACTL.SCIS and PM1_CNT.SCI_EN registers)	3
IRQ12	Slave IRQ4	• SERIRQ (12), or • PIRQx	3

Table 29-1. 8259 PIC Input Mapping (Sheet 2 of 2)

I/O PIC Input	Master or Slave 8259 PIC Input	Interrupts Routed to This PIC Input	Note
IRQ13	Slave IRQ5	• GPIO	
IRQ14	Slave IRQ6	• SERIRQ (14), or • IRQ15 from ISA IDE Interrupt, or • PIRQx	3
IRQ15	Slave IRQ7	• SERIRQ (15), or • PIRQx	3

Notes:
1. Interrupts can individually be programmed to be edge or level, except for IRQ0, IRQ2, and IRQ8#.
2. The slave 8259 controller is cascaded onto the master 8259 controller through the master controller interrupt input IRQ2.
3. For routing of the PIRQA through PIRQH interrupts, see Table 6-1, "PIRQA through PIRQH Routing Register IRQ Decode" on page 117.

The SoC cascades the slave controller onto the master controller through the master controller interrupt input 2. This means only 15 interrupts exist for the SoC PIC.

Note: Active-low interrupt sources (such as PIRQ#) are inverted inside the SoC. In the following descriptions of the 8259s, interrupt levels are in reference to signals at the internal interface of the 8259s after the required inversions have occurred. Therefore, the term high indicates active, which means low on an originating PIRQ#.

29.2.1 Interrupt Handling

29.2.1.1 Generating Interrupts

The PIC interrupt sequence involves 3 bits, from the IRR, ISR, and IMR, for each interrupt level. These bits are used to determine the interrupt vector returned and the status of any other pending interrupts. Table 29-2 defines the IRR, ISR, and IMR.

Table 29-2. Interrupt Status Registers

Bit	Description
IRR	**Interrupt Request Register:** This bit is set on a low-to-high transition of the interrupt line in the edge mode and by an active high level in the level mode.
ISR	**Interrupt Service Register:** This bit is set, and the corresponding IRR bit cleared, when an interrupt acknowledge cycle is seen, and the vector returned is for that interrupt.
IMR	**Interrupt Mask Register:** This bit determines whether an interrupt is masked. Masked interrupts do not generate INTR.

29.2.1.2 Acknowledging Interrupts

The processor generates an interrupt acknowledge cycle that is translated into a interrupt acknowledge cycle to the SoC. The PIC translates this command into two internal INTA# pulses expected by the 8259 controllers. The PIC uses the first internal INTA# pulse to freeze the state of the interrupts for priority resolution. On the second INTA# pulse, the master or slave sends the interrupt vector to the processor with the acknowledged interrupt code. This code is based on the ICW2.IVBA bits, combined with the ICW2.IRL bits representing the interrupt within that controller.

References to the ICWx and OCWx registers in Table 29-3 are relevant to both the master and slave 8259 controllers.

Table 29-3. Content of Interrupt Vector Byte

Master, Slave Interrupt	Bits [7:3]	Bits [2:0]
IRQ7,15	ICW2.IVBA	111
IRQ6,14		110
IRQ5,13		101
IRQ4,12		100
IRQ3,11		011
IRQ2,10		010
IRQ1,9		001
IRQ0,8		000

29.2.1.3 Hardware/Software Interrupt Sequence

1. One or more of the Interrupt Request lines (IRQ) are raised high in the edge mode or seen high in the level mode, setting the corresponding IRR bit.
2. The PIC sends INTR active to the processor if an asserted interrupt is not masked.
3. The processor acknowledges the INTR and responds with an interrupt acknowledge cycle.
4. When observing the special cycle, the SoC converts it into the two cycles that the internal 8259 pair responds. Each cycle appears as an interrupt acknowledge pulse on the internal INTA# pin of the cascaded interrupt controllers.
5. When receiving the first internally generated INTA# pulse, the highest priority ISR bit is set and the corresponding IRR bit is reset. On the trailing edge of the first pulse, a slave identification code is broadcast by the master to the slave on a private, internal 3-bit wide bus. The slave controller uses these bits to determine if it must respond with an interrupt vector during the second INTA# pulse.
6. When receiving the second internally generated INTA# pulse, the PIC returns the interrupt vector. If no interrupt request is present because the request was too short in duration, the PIC returns vector 7 from the master controller.
7. This completes the interrupt cycle. In AEOI mode, the ISR bit is reset at the end of the second INTA# pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

29.2.2 Initialization Command Words (ICWx)

Before an operation begins, each 8259 must be initialized. In the SoC, this is a 4-byte sequence. The four initialization command words are referred to by their acronyms: ICW1, ICW2, ICW3, and ICW4.

The base address for each 8259 initialization command word is a fixed location in the I/O space: 20h for the master controller and A0h for the slave controller.

29.2.2.1 ICW1

A write to the master or slave controller base address with data bit 4 equal to 1 is interpreted as a write to ICW1. Upon sensing this write, the PIC expects three more byte writes to 21h for the master controller or A1h for the slave controller to complete the ICW sequence.

A write to ICW1 starts the initialization sequence during which the following automatically occurs:

1. Following initialization, an Interrupt Request (IRQ) input must make a low-to-high transition to generate an interrupt.
2. The Interrupt Mask Register is cleared.
3. IRQ7 input is assigned priority 7.
4. The slave mode address is set to 7.
5. Special mask mode is cleared, and the status read is set to IRR.

29.2.2.2 ICW2

The second write in the sequence (ICW2) is programmed to provide bits [7:3] of the interrupt vector that are released during an interrupt acknowledge. A different base is selected for each interrupt controller.

29.2.2.3 ICW3

The third write in the sequence (ICW3) has a different meaning for each controller.

- For the master controller, ICW3 indicates which IRQ input line cascades the slave controller. Within the SoC, IRQ2 is used. Therefore, MICW3.CCC is set to a 1 and the other bits are set to 0s.
- For the slave controller, ICW3 is the slave identification code used during an interrupt acknowledge cycle. On interrupt acknowledge cycles, the master controller broadcasts a code to the slave controller if the cascaded interrupt won arbitration on the master controller. The slave controller compares this identification code to the value stored in its ICW3, and if it matches, the slave controller assumes responsibility for broadcasting the interrupt vector.

29.2.2.4 ICW4

The final write in the sequence (ICW4) must be programmed for both controllers. At least, ICW4.MM must be set to a 1 to indicate the controllers are operating in an Intel® architecture system.

29.2.3 Operation Command Words (OCW)

These command words reprogram the interrupt controller to operate in various interrupt modes:

- OCW1 masks and unmasks the interrupt lines.
- OCW2 controls the rotation of interrupt priorities when in rotating priority mode and controls the EOI function.
- OCW3 sets up the ISR/IRR reads, enables/disables the Special Mask Mode (SMM), and enables/disables the polled interrupt mode.

29.3 Operation

29.3.1 Fully-Nested Mode

In this mode, interrupt requests are ordered in priority from 0 through 7, with 0 being the highest. When an interrupt is acknowledged, the highest priority request is determined and its vector placed on the bus. Additionally, the ISR for the interrupt is set. This ISR bit remains set until: the processor issues an EOI command immediately before returning from the service routine or if in AEOI mode, on the trailing edge of the second INTA#. While the ISR bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels generate another interrupt. Interrupt priorities are changed in the rotating priority mode.

29.3.2 Special Fully-Nested Mode

This mode is used in the case of a system where cascading is used, and the priority has to be conserved within each slave. In this case, the special fully-nested mode is programmed to the master controller. This mode is similar to the fully-nested mode with the following exceptions:

- When an interrupt request from a certain slave is in service, this slave is not locked out from the master priority logic, and further interrupt requests from higher priority interrupts within the slave are recognized by the master and initiate the interrupts to the processor. In the normal-nested mode, a slave is masked out when its request is in service.
- When exiting the interrupt service routine, the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a Non-specific EOI command to the slave and then reading its ISR. If it is 0, a non-specific EOI is also sent to the master.

29.3.3 Automatic Rotation Mode (Equal Priority Devices)

In some applications, there are a number of interrupting devices of equal priority. The automatic rotation mode provides for a sequential eight-way rotation. In this mode, a device receives the lowest priority after being serviced. In the worst case, a device requesting an interrupt has to wait until each of the seven other devices are serviced at most once.

Two ways to accomplish automatic rotation using OCW2.REOI are: the rotation on the Non-specific EOI command (OCW2.REOI=101b) and the rotation in the automatic EOI mode which is set by (OCW2.REOI=100b).

29.3.4 Specific Rotation Mode (Specific Priority)

The software changes the interrupt priorities by programming the bottom priority. For example, if IRQ5 is programmed as the bottom priority device, then IRQ6 is the highest priority device. The Set Priority command is issued in OCW2 to accomplish this, where OCW2.REOI=11xb, and OCW2.ILS is the binary priority level code of the bottom priority device.

In this mode, the internal status is updated by the software control during OCW2. However, it is independent of the EOI command. Priority changes are executed during an EOI command by using the Rotate-on-Specific EOI command in OCW2 (OCW2.REOI=111b) and OCW2.ILS=IRQ level to receive bottom priority.

29.3.5 Poll Mode

The poll mode conserves space in the interrupt vector table. Multiple interrupts that are serviced by one interrupt service routine do not need separate vectors if the service routine uses the Poll command. The poll mode is also used to expand the number of interrupts. The polling interrupt service routine calls the appropriate service routine instead of providing the interrupt vectors in the vector table. In this mode, the INTR output is not used and the microprocessor internal interrupt enable flip-flop is reset, disabling its interrupt input. Service to the devices is achieved by the software using a Poll command.

The Poll command is issued by setting OCW3.PMC. The PIC treats its next I/O read as an interrupt acknowledge, sets the appropriate ISR bit if there is a request, and reads the priority level. Interrupts are frozen from the OCW3 write to the I/O read. The byte returned during the I/O read contains a 1 in bit 7 if there is an interrupt and the binary code of the highest priority level in bits [2:0].

29.3.6 Edge- and Level-Triggered Mode

In ISA systems, this mode is programmed using ICW1.LTIM, which sets level or edge for the entire controller. In the SoC, this bit is disabled and a register for edge- and level-triggered mode selection per interrupt input is included. This is the Edge/Level Control Registers ELCR1 and ELCR2.

If the ELCR bit is 0, an interrupt request is recognized by a low-to-high transition on the corresponding IRQ input. The IRQ input remains high without generating another interrupt. If the ELCR bit is 1, an interrupt request is recognized by a high level on the corresponding IRQ input, and an edge detection is not needed. The interrupt request must be removed before the EOI command is issued to prevent a second interrupt from occurring.

In both the edge- and level-triggered modes, the IRQ inputs must remain active until after the falling edge of the first internal INTA#. If the IRQ input goes inactive before this time, a default IRQ7 vector is returned.

29.3.7 End of Interrupt (EOI) Operations

An EOI occurs in one of two fashions: by a command word write issued to the PIC before returning from a service routine, the EOI command; or automatically when the ICW4.AEOI bit is set to 1.

29.3.8 Normal End of Interrupt

In normal EOI, the software writes an EOI command before leaving the interrupt service routine to mark the interrupt as completed. The two forms of EOI commands are: specific and non-specific. When a Non-specific EOI command is issued, the PIC clears the highest ISR bit of those that are set to 1. A non-specific EOI is the normal mode of operation of the PIC within the SoC, as the interrupt being serviced currently is the interrupt entered with the interrupt acknowledge. When the PIC is operated in the modes that preserve the fully-nested structure, the software determines which ISR bit to clear by issuing a specific EOI.

An ISR bit that is masked is not cleared by a non-specific EOI if the PIC is in the special mask mode. An EOI command must be issued for both the master and slave controller.

29.3.9 Automatic End of Interrupt Mode

In this mode, the PIC automatically performs a non-specific EOI operation at the trailing edge of the last interrupt acknowledge pulse. From a system standpoint, this mode is used only when a nested multi-level interrupt structure is not required within a single PIC. The AEOI mode is only used in the master controller and not the slave controller.

Note: Both the master and slave PICs have an AEOI bit: MICW4.AEOI and SICW4.AEOI, respectively. Only the MICW4.AEOI bit is set by the software. The SICW4.AEOI bit is not set by the software.

29.3.10 Masking Interrupts

29.3.10.1 Masking on an Individual Interrupt Request

Each interrupt request is masked individually by the Interrupt Mask Register (IMR). This register is programmed through OCW1. Each bit in the IMR masks one interrupt channel. Masking IRQ2 on the master controller masks all requests for service from the slave controller.

29.3.10.2 Special Mask Mode

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

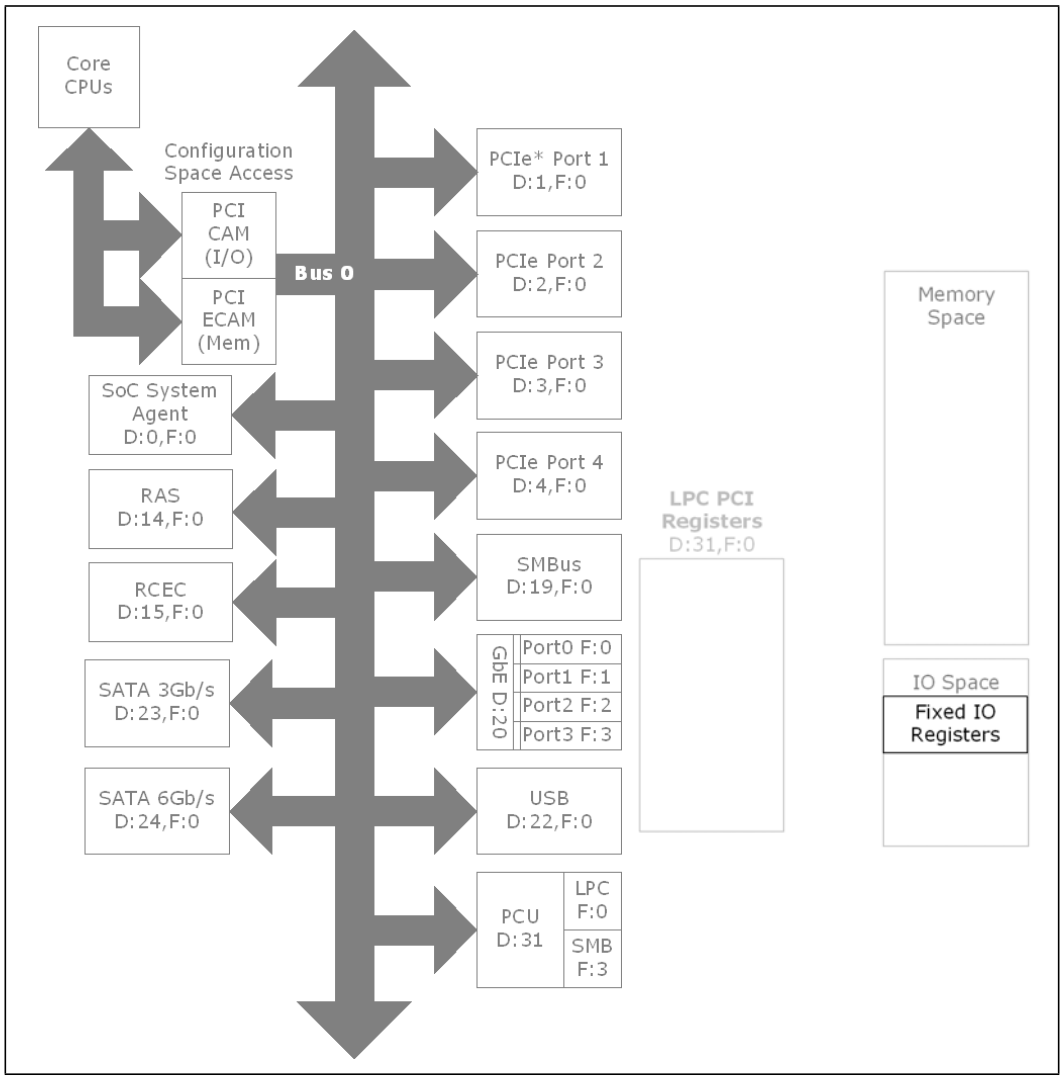
The special mask mode enables all interrupts not masked by a bit set in the Mask Register. Normally, when an interrupt service routine acknowledges an interrupt without issuing an EOI to clear the ISR bit, the interrupt controller inhibits all lower priority requests. In the special mask mode, any interrupts are selectively enabled by loading the mask register with the appropriate pattern.

The special mask mode is set by OCW3.ESMM=1b and OCW3.SMM=1b, and cleared where OCW3.ESMM=1b and OCW3.SMM=0b.

29.4 Register Map

Figure 29-2 shows the SoC 8529 registers from a system viewpoint.

Figure 29-2. 8259 PIC Register Map



29.4.1 I/O Mapped Registers

The interrupt controller registers are located at 20h and 21h for the master controller (IRQ0 - 7) and at A0h and A1h for the slave controller (IRQ8 - 13). These registers have multiple functions, depending upon the data written to them. Table 29-4 is a description of the different register possibilities for each address.

Note: The register descriptions after Table 29-4 represent one register possibility.

Table 29-4. I/O Registers Alias Locations

Registers	Original I/O Location	Alias I/O Locations
MICW1 MOCW2 MOCW3	20h	24h
		28h
		2Ch
		30h
		34h
		38h
		3Ch
MICW2 MICW3 MICW4 MOCW1	21h	25h
		29h
		2Dh
		31h
		35h
		39h
		3Dh
SICW1 SOCW2 SOCW3	A0h	A4h
		A8h
		ACh
		B0h
		B4h
		B8h
		BCh
SICW2 SICW3 SICW4 SOCW1	A1h	A5h
		A9h
		ADh
		B1h
		B5h
		B9h
		BDh
ELCR1	4D0h	N/A
ELCR2	4D1h	N/A

The register descriptions in Table 29-5 are one set of fixed I/O addresses to access the 8254 I/O registers. Table 29-4 shows the original I/O address and the associated aliased I/O addresses.

Table 29-5. 8259 I/O Registers in Fixed I/O Space (One Possibility)

Offset	Name	Description
0x020	MICW1	Master Initialization Command Word 1
0x021	MICW2	Master Initialization Command Word 2
0x024	MOCW2	Master Operational Control Word 2
0x025	MICW3	Master Initialization Command Word 3
0x028	MOCW3	Master Operational Control Word 3
0x029	MICW4	Master Initialization Command Word 4
0x02D	MOCW1	Master Operational Control Word 1 (Interrupt Mask)
0x0A0	SICW1	Slave Initialization Command Word 1
0x0A1	SICW2	Slave Initialization Command Word 2
0x0A4	SOCW2	Slave Operational Control Word 2
0x0A5	SICW3	Slave Initialization Command Word 3
0x0A8	SOCW3	Slave Operational Control Word 3
0x0A9	SICW4	Slave Initialization Command Word 4
0x0AD	SOCW1	Slave Operational Control Word 1 (Interrupt Mask)
0x4D0	ELCR1	Master Edge/Level Control
0x4D1	ELCR2	Slave Edge/Level Control

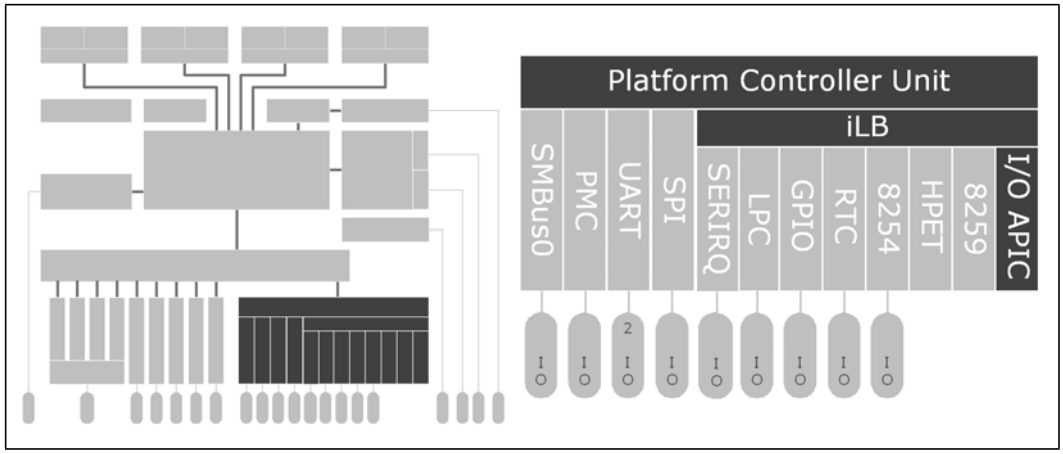
§ §

30 I/O Advanced APIC (I/O APIC)

The SoC contains an integrated I/O Advanced Programmable Interrupt Controller (I/O APIC). The I/O APIC priority schemata and advanced Interrupt Request (IRQ) management are more complex than the 8259 Programmable Interrupt Controller (PIC).

Line interrupts from multiple sources, including legacy devices, are sent to the I/O APIC. These interrupts originate from the SoC interrupt decoder, the SoC serial interrupt decoder, and the interrupt router in the integrated legacy block. These line-based interrupts are sent to the CPU local APIC.

Figure 30-1. I/O APIC Covered in This Chapter



30.1 Signal Descriptions

The I/O APIC has no external signal pins.

30.2 Features

- 24 interrupt lines
  - — IRQ0-23
- Edge- or level-trigger mode per interrupt
- Active low or high polarity per interrupt
- MSIs target the specific processor core
- Established APIC programming model

30.3 Architectural Overview

There are 24 I/O Redirection Table Entry registers. Each register is a dedicated entry for each interrupt input signal. For information about interrupts routed to the I/O APIC, see Chapter 6, "Interrupt Architecture."

Unlike IRQ pins of the 8259A, the notion of interrupt priority is completely unrelated to the position of the physical interrupt input signal on the I/O APIC. Instead, the software determines the vector (and therefore the priority) for each corresponding interrupt input signal. For each interrupt signal, the operating system also specifies the signal polarity (low active or high active), whether the interrupt is signaled as edges or levels, as well as the destination and delivery mode of the interrupt.

The information in the redirection table translates the corresponding interrupt pin information into an inter-APIC message.

Table 30-1. I/O APIC Internal Registers

Offset	Symbol	Register
00h	ID	Identification. The software must program an APIC Identification (AID) value before using the APIC.
01h	VS	Version. A read-only register identifying it as IOxAPIC with 24 I/O interrupts.
02h through 0Fh	-	Reserved
10h and 11h	RTE0	Redirection Table Entry 0
12h and 13h	RTE1	Redirection Table Entry 1
…	…	…
3Eh and 3Fh	RTE23	Redirection Table Entry 23
40h through FFh	-	Reserved

The software does not attempt to write to reserved registers. Reserved registers may return non-zero values when read.

See Section 30.3.3 for descriptions of these I/O APIC internal registers.

30.3.1 APIC ID and Version Registers

The I/O APIC has a 32-bit APIC Identification register and a 32-bit Version register.

30.3.2 Interrupt Redirection Registers

The SoC I/O APIC accommodates up to 24 I/O interrupts. It provides a 64-bit I/O-Interrupt Redirection register for each. The Redirection register contains:

- An 8-bit Destination ID (DID) of the local APIC for the interrupt.
- An 8-bit Extended Destination ID (DID) of the local APIC for the interrupt.
- 4 bits to indicate the mask, trigger mode, remote IRR (for trigger mode), and polarity for the interrupt.
- 5 bits for the interrupt delivery mode and delivery status.
- 8 bits containing the interrupt Vector (VCT) with values of 10h through FEh.

The MSIs generated by the I/O APIC are sent as 32-bit memory writes to the local APIC. The Destination ID (DID) and Extended Destination ID (EDID) are used to target a specific processor core local APIC.

30.3.3 Accessing the I/O APIC Internal Registers

The I/O APIC internal registers are accessed indirectly. They are accessed using three registers in the memory space. The three registers have fixed memory addresses as shown in Table 30-2.

Table 30-2. I/O APIC Register Access and EOI Register

Fixed Address in the Memory Space	Default	Name	Description
0xFEC00000	00h	IOAPIC_IDX	IDX - Index register
0xFEC00010	0000_0000h	IOAPIC_WDW	WDW - Window register
0xFEC00040	0000_0000h	IOAPIC_EOI	EOI - End Of Interrupt (EOI) register

This 8-bit Index register (IDX) selects which of the 256 indirect registers, 00h through FFh, appears in the Window register (WDW) so it is manipulated by the software. In other words, the software programs the 8-bit IDX register to select one of the 256, 32-bit APIC internal registers. The 32-bit selected register appears in the 32-bit WDW.

- The 32-bit APIC Identification register is accessed through this mechanism.
- The 32-bit Version register is accessed through this mechanism.
- The twenty-four, 64-bit Redirection Table Entries (RTE0 through RTE23) are only accessed 32 bits at a time; one Dword per IDX value.
- Some of the 256 registers are reserved. See Table 30-1.

The registers that appear in the WDW register are described in the following sections and tables.

30.3.3.1 Identification (ID) Register

The 32-bit Identification (ID) register is accessed at offset 00h.

Table 30-3. Identification (ID) Register

Bits	Type	Reset	Description
31:28	RO	0	*Reserved*
27:24	RW	0	**APIC Identification (AID):** Software must program this value before using the APIC.
23:16	RO	0	*Reserved*
15	RW	0	Scratchpad
14	RW	0	*Reserved. Writing to this bit has no effect.*
13:0	RO	0	*Reserved*

30.3.3.2 Version (VS) Register

The 32-bit Version (VS) register is accessed at offset 01h.

Table 30-4. Version (VS) Register

Bits	Type	Reset	Description
31:24	RO	0	*Reserved*
23:16	RO	17h	**Maximum Redirection Entries (MRE):** This is the entry number (0 being the lowest entry) of the highest entry in the redirection table. This field is hard-wired to 17h to indicate 24 interrupts.
15	RO	0	**Pin Assertion Register Supported (PRQ):** The IOxAPIC does not implement the Pin Assertion Register.
14:8	RO	0	*Reserved*
7:0	RO	20h	**Version (VS):** Identifies the implementation version as IOxAPIC.

30.3.3.3 Redirection Table Entry (RTE[23:0]) Registers

All 24 of the 64-bit Redirection Table Entry (RTE) Registers have the same format which is shown below. Each 64-bit register is accessed as two, 32-bit register accesses starting at offsets 10-11h (RTE[0]). The RTE[23] register is accessed at offsets 3E-3Fh.

Table 30-5. Redirection Table Entry (RTE[23:0]) Registers

Bits	Type	Reset	Description
63:56	RW	X	**Destination ID (DID):** Destination ID of the local APIC.
55:48	RW	X	**Extended Destination ID (EDID):** Extended destination ID of the local APIC.
47:17	RO	0	*Reserved*
16	RW	1	**Mask (MSK):** When set, interrupts are not delivered nor held pending. When cleared, and edge or level on this interrupt results in the delivery of the interrupt.
15	RW	X	**Trigger Mode (TM):** When cleared, the interrupt is edge sensitive. When set, the interrupt is level sensitive.
14	RW	X	**Remote IRR (RIRR):** This is used for level triggered interrupts its meaning is undefined for edge triggered interrupts. This bit is set when IOxAPIC sends the level interrupt message to the CPU. This bit is cleared when an EOI message is received that matches the VCT field. This bit is never set for SMI, NMI, INIT, or ExtINT delivery modes.
13	RW	X	**Polarity (POL):** This specifies the polarity of each interrupt input. When cleared, the signal is active high. When set, the signal is active low.
12	RO	X	**Delivery Status (DS):** This field contains the current status of the delivery of this interrupt. When set, an interrupt is pending and not yet delivered. When cleared, there is no activity for this entry.
11	RW	X	**Destination Mode (DSM):** This field is used by the local APIC to determine whether it is the destination of the message.
10:8	RW	X	**Delivery Mode (DLM):** This field specifies how the APICs listed in the destination field should act upon reception of this signal. Certain Delivery Modes only operate as intended when used in conjunction with a specific trigger mode. The encoding is: 000 Fixed 001 Lowest Priority 010 SMI – Not supported 011 *Reserved* 100 NMI – Not supported 101 INIT – Not supported 110 *Reserved* 111 ExtINT
7:0	RW	X	**Vector (VCT):** This field contains the interrupt vector for this interrupt. Values range between 10h and FEh.

30.3.4 End Of Interrupt (EOI) Register

The 32-bit I/O APIC EOI register is in the memory space at address FEC0_0040h. See Table 30-2. The register is written by the local APIC in the processor.

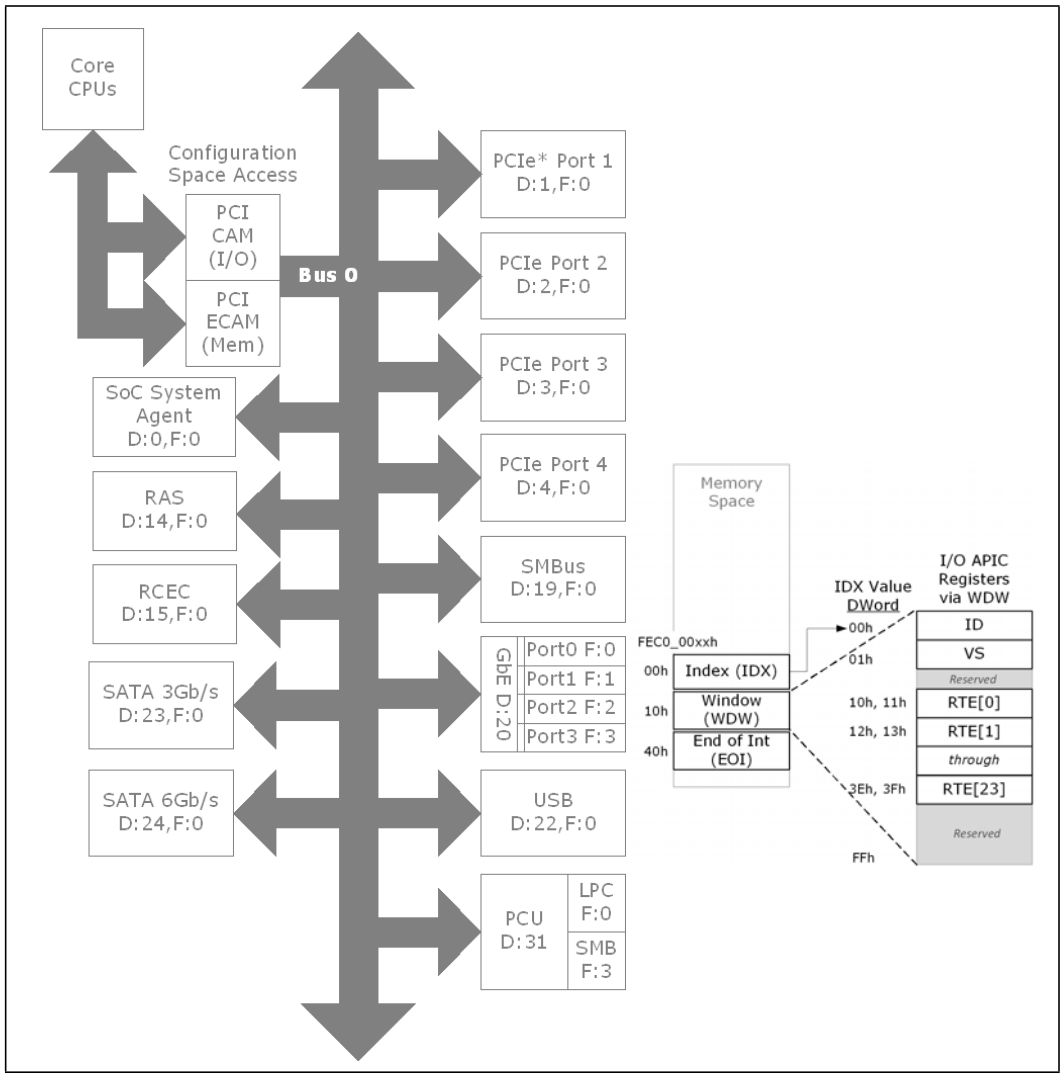
When a memory write is issued to the 32-bit End Of Interrupt (EOI) register, the I/O APIC checks the lower 8 bits written to this register and compares them with the Vector (VCT) field for each of the 24 Redirection Table Entries (RTE 0 through RTE 23) in the I/O Redirection Table.

When a match is found, the Remote IRR (RIRR) bit for that entry is cleared. If multiple entries have the same vector, each of those entries have its RIRR bit cleared.

30.4 Register Map

Figure 30-2 shows the SoC I/O APIC registers from a system viewpoint.

Figure 30-2. I/O APIC Register Map



30.4.1 Memory-Mapped Registers

The three memory-mapped I/O APIC registers have fixed memory addresses and are shown in Table 30-2. The I/O APIC controller registers to which they allow access, are listed in Table 30-1.

§ §

Volume 3: Electrical, Mechanical, and Thermal

31 Signal Names and Descriptions

31.1 Overview

This chapter provides a detailed description of the signals and bootstrap definitions for the SoC. The signals are arranged in functional groups according to their associated interface.

Each signal description table has the following headings:

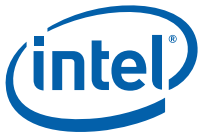
- **Signal:** The name of the signal ball/pin.
- **Description:** A brief explanation of the signal function.
- **Power Rail:** Power rails used to supply power to the I/O signal are defined in Table 31-1.

Table 31-1. Buffer Power Rails

<table>
<tr><th>Power Rail</th><th>Description</th></tr>
<tr><td>V1P0A</td><td>1.00V Suspend (SUS) rail. This rail is active in the S5 power state.</td></tr>
<tr><td>V1P0S</td><td>1.00V Core rail. This rail is inactive in the S5 power state, but powered in S0.</td></tr>
<tr><td>V3P3A</td><td>3.3V SUS rail. This rail is active in the S5 power state.</td></tr>
<tr><td>V3P3S</td><td>3.3V Core rail. This rail is inactive in the S5 power state, but powered in S0.</td></tr>
<tr><td>VRTC3P0</td><td>3.3V RTC Power rail</td></tr>
<tr><td>VDDQ</td><td>DDR3 I/O Voltage (1.5V/1.35V)

Note: Using VDDQA and VDDQB instead of a single VDDQ is based on the DIMM topology. When the DIMMs are on either side of the SoC, VDDQA and VDDQB are used for Channel 0 and Channel 1 respectively.
<table>
<tr><th>Technology</th><th>Voltage (VDDR)</th><th>Speeds (MT/s)</th></tr>
<tr><td>DDR3</td><td>1.50V</td><td>1333, 1600</td></tr>
<tr><td>DDR3L</td><td>1.35V</td><td>1333, 1600</td></tr>
</table>
</td></tr>
</table>

Note: Refer to Table 34-3, "Voltage Supply Requirements Under Normal Operating Conditions" on page 690 for more details on the voltage rails.



- **Direction and Type:** The buffer direction and type.
  - Buffer direction is input (I), output (O), or bi-directional (I/O or IO).
  - Buffer Types are defined in Table 31-2.

Table 31-2. Buffer Types

Buffer Type	Buffer Description
CMOS_V1P0	1.00V CMOS Buffer: Buffer Technology = CMOS Buffer Power Well = V1P0S or V1P0A
CMOS_V1P0_OD	1.00V CMOS Open Drain Buffer: Buffer Technology = CMOS Buffer Power Well = V1P0S or V1P0A
CMOS_V3P3	3.3V CMOS Buffer: Buffer Technology = CMOS Buffer Power Well = V3P3S or V3P3A
CMOS_V3P3_OD	3.3V CMOS Open Drain Buffer: Buffer Technology = CMOS Buffer Power Well = V3P3S or V3P3A
DDR	DDR3 Buffer: Buffer Technology = CMOS Buffer Power Well = VDDR ***Note:*** The applicable DDR3 Buffer power (VDDQ) is defined in Table 31-1
LV DIFF	Low-Voltage Differential I/O Buffers
Analog	Analog reference voltage, input/output signal, or connection to an external or internal passive component.

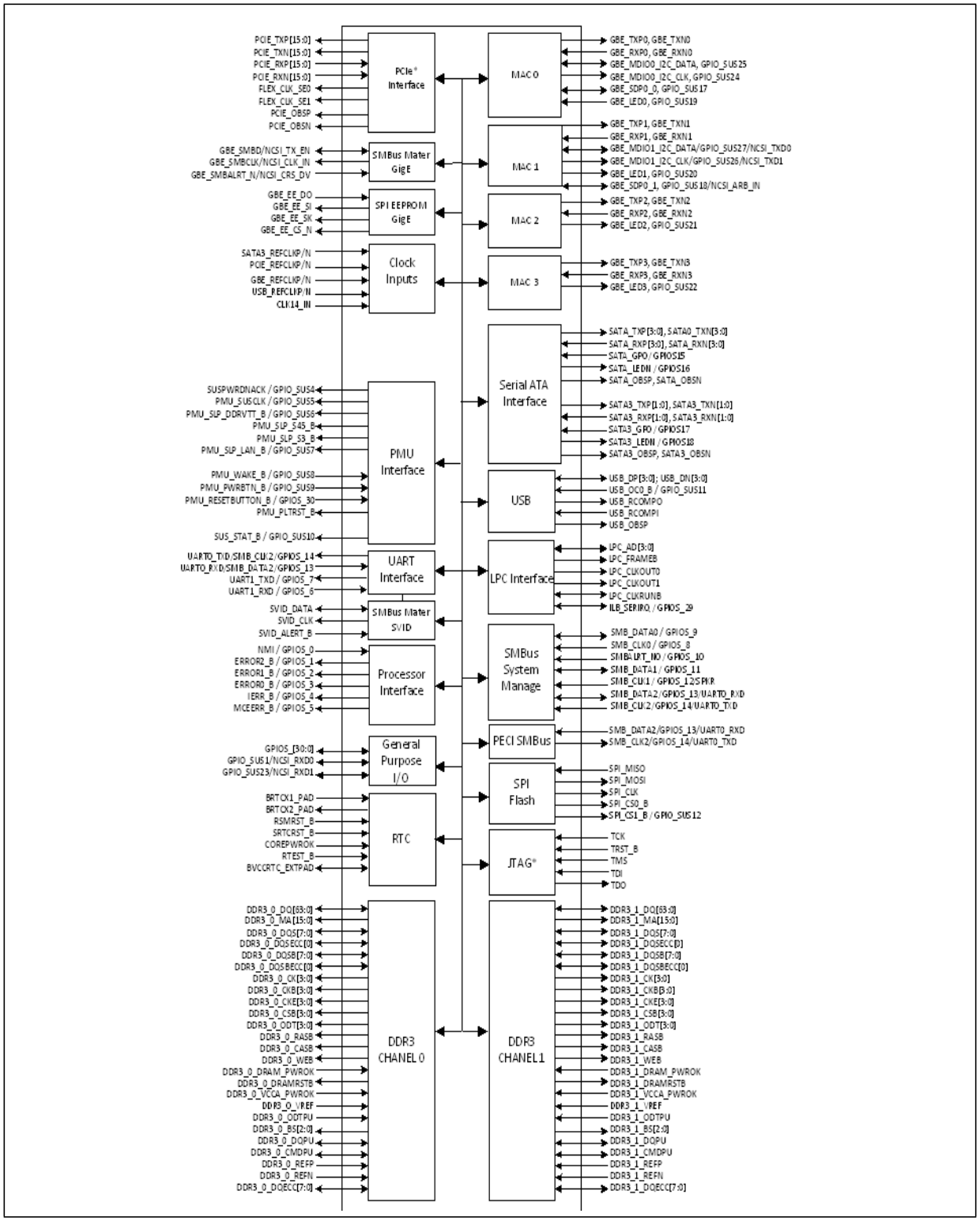
31.2 Name Convention

Table 31-3 provides the legend for interpreting the I/O Type field that appears the tables in this section.

Table 31-3. Signal Type Definitions

Type	Description
# or _B or _N	Active low signal
CMOS	CMOS buffers
DDR	Double Data Rate
I	Input pin
I/O or IO	Bi-directional Input /Output pin
I/OD	Bi-directional Input/Open Drain output pin
LV DIFF	Low-Voltage Differential
LVCMOS	Low-Voltage Complementary Metal Oxide Semiconductor
LVTTL	Low-Voltage Transistor-Transistor Logic
NC	No Connection to pin
O	Output pin
OD	Open Drain output pin
RSVD	Reserved Pin. This signal must be connected as described in signal description.
T/S	Tri-State pin

Figure 31-1. Interface Signals Block Diagram



31.3 System DDR Memory Signals

Table 31-4. DDR0 Signals (Sheet 1 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
DDR3_0_DQ[63:0]	I/O	DDR	64			VDDQ	DDR3 Data Bus: Memory read and write data. Data signal interface to the SDRAM data bus. These 64-bit signals have 8-byte lanes, and each byte lane has a corresponding strobe pair.
DDR3_0_MA[15:0]	O	DDR	16			VDDQ	DDR3 Memory Address: Provides multiplexed row and column address to memory. Provides the row address for active commands and the column address and Auto-Pre-charge bit for read/write commands to select one location out of the memory array in the respective bank. A10 is sampled during a Pre-charge command to determine whether the Pre-charge applies to one bank (A10 LOW) or all banks (A10 HIGH). If only one bank is to be pre-charged, the bank is selected by BA0 - BA2. The address inputs also provide the op-code during MRS or EMRS commands.
DDR3_0_DQS[7:0]	I/O	DDR	8			VDDQ	DDR3 Data Strobes: During writes, driven by the CDV offset so as to be centered in the data phase. During reads, driven by memory devices edge-aligned with data. The following list matches the data strobe with the data bytes: (DQS_7: DQ[63:56] …. DQS_0: DQ[7:0]). The data strobes may be used in single-ended mode or paired with optional complementary signals DQS_B to provide differential-pair signaling to the system during both reads and writes. A control bit at EMR(1)[A10] enables or disables all complementary data strobe signals.
DDR3_0_DQSECC[0]	I/O	DDR	1			VDDQ	DDR3 ECC Strobe: Differential-pair output with read-data ECC, differential-pair input with write-data ECC. Edge-aligned with read-data ECC, centered in write-data ECC.

Table 31-4. DDR0 Signals (Sheet 2 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
DDR3_0_DQSB[7:0]	I/O	DDR	8			VDDQ	DDR3 Data Strobes: During writes, driven by CDV offset so as to be centered in the data phase. During reads, driven by memory devices edge-aligned with data. The following list matches the data strobe with the data bytes: (DQS_7: DQ[63:56] …. DQS_0: DQ[7:0]). The data strobes may be used in single-ended mode or paired with optional complementary signals DQS_B to provide differential-pair signaling to the system during both reads and writes. A control bit at EMR(1)[A10] enables or disables all complementary data strobe signals.
DDR3_0_DQSBECC[0]	I/O	DDR	1			VDDQ	DDR3 ECC Strobe: Differential-pair output with read-data ECC, differential-pair input with write-data ECC. Edge-aligned with read-data ECC, centered in write-data ECC.
DDR3_0_CK[3:0]	O	DDR	4			VDDQ	DDR3 Differential Clock: All address and control input signals are sampled on the crossing of the positive edge of CK and negative edge of CKB. Output (read) data is referenced to the crossings of CK and CKB (both directions of crossing).
DDR3_0_CKB[3:0]	O	DDR	4			VDDQ	DDR3 Differential Clock: All address and control input signals are sampled on the crossing of the positive edge of CK and negative edge of CKB. Output (read) data is referenced to the crossings of CK and CKB (both directions of crossing).

Table 31-4. DDR0 Signals (Sheet 3 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
DDR3_0_CKE[3:0]	O	DDR	4			VDDQ	DDR3 Clock Enable: (active high). CKE is used for power control of the DRAM devices. For the DRAM Devices: CKE HIGH activates, and CKE LOW deactivates, internal clock signals and device input buffers and output drivers. Taking CKE LOW provides Pre-charge Power Down and Self-Refresh operation (all banks idle) or Active Power Down (row Active in any bank). CKE is synchronous for a power down entry and exit, and for self-refresh entry. CKE is asynchronous for self-refresh exit. After VREF has become stable during the power-on and initialization sequence, it must be maintained for proper operation of the CKE receiver. For proper self-refresh entry and exit, VREF must be maintained to this input. CKE must be maintained HIGH throughout read and write accesses. Input buffers, excluding CK, CKB, ODT, and CKE are disabled during power down. Input buffers, excluding CKE, are disabled during self-refresh.
DDR3_0_CSB[3:0]	O	DDR	4			VDDQ	DDR3 Chip Select: (active low). These signals determine whether a command is valid in a given cycle for the devices connected to it. All commands are masked when CSB is registered HIGH. CSB provides for external Rank selection on systems with multiple Ranks. CSB is considered part of the command code.

Table 31-4. DDR0 Signals (Sheet 4 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
DDR3_0_ODT[3:0]	O	DDR	4			VDDQ	DDR3 On-Die Termination Enable: (active high). ODT (registered HIGH) enables termination resistance internal to the DDR device SDRAM. When the ODT feature is enabled, it is dynamically enabled for the receiver of the data. The SoC does this internally for read data returning from the DRAM devices. For write data to the DRAM devices, the M_ODT[] pins are asserted to enable ODT within the DRAM devices themselves. Because ODT consumes power, when the feature is enabled, it is control dynamically by the SoC. ODT impacts the DQ, DQS, and DM signals. The ODT pin is ignored by the DDR devices if the EMR(1) is programmed to disable ODT. One pin per rank.
DDR3_0_RASB	O	DDR	1			VDDQ	DDR3 Row Address Strobe: (active low). Used with CASB and RASB (along with CSB) to define commands. RAS, CAS, and WE (along with CS) define the command being entered.
DDR3_0_CASB	O	DDR	1			VDDQ	DDR3 Column Address Strobe: (active low). Used with CAS#, RAS#, and CS# to define commands. RAS, CAS, and WE (along with CS) define the command being entered.
DDR3_0_WEB	O	DDR	1			VDDQ	DDR3 Write Enable: (active low). Used with CAS#, RAS#, and CS# to define commands. RAS, CAS, and WE (along with CS) define the command being entered.
DDR3_0_DRAM_PWROK	I	DDR	1			VDDQ	DDR3 DRAM POWER OK. Active high signal indicates that the DDR PHY voltage (VDDR) is good.
DDR3_0_DRAMRSTB	O	DDR	1			VDDQ	DDR3 DRAM Reset: (active low). Asynchronous output reset signal to DIMM and SDRAM devices. It is common to all ranks.
DDR3_0_VCCA_PWROK	I	DDR	1			VDDQ	DDR3 Indication to the DDRIO that core well voltage is valid. This is connected to the SoC input COREPWROK (except they are different voltages).
DDR3_0_VREF	I	DDR	1		100 1%, PD	VDDQ	External resistor for internal voltage divider.
DDR3_0_ODTPU	I/O	DDR	1			VDDQ	DDR3 Compensation Pad. Board trace + External Precision resistor. The resistor is pulled-down to VSS.

Table 31-4. DDR0 Signals (Sheet 5 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
DDR3_0_BS[2:0]	O	DDR	3			VDDQ	DDR3 Bank Select: Defines which banks are being addressed within each rank. BA0 - BA2 define to which bank an Active, Read, Write or Pre-charge command is being applied (for 256 MB and 512 MB, BA2 is not applied). Bank address also determines if the mode register or one of the extended mode registers is to be accessed during a MRS or EMRS command cycle.
DDR3_0_DQPU	I/O	DDR	1			VDDQ	DDR3 Compensation Pad. Board trace + External Precision resistor. The resistor is pulled-down to VSS.
DDR3_0_CMDPU	I/O	DDR	1			VDDQ	DDR3 Compensation Pad. Board trace + External Precision resistor. The resistor is pulled-down to VSS.
DDR3_0_MON1P	I/O	DDR	1			VDDQ	DDR3 PLL Monitor Port1.
DDR3_0_MON1N	I/O	DDR	1			VDDQ	DDR3 PLL Monitor Port1.
DDR3_0_MON2P	I/O	DDR	1			VDDQ	DDR3 PLL Monitor Port2.
DDR3_0_MON2N	I/O	DDR	1			VDDQ	DDR3 PLL Monitor Port2.
DDR3_0_REFP	I	DDR	1			VDDQ	DDR3 Clock Reference: Differential-pair input. Used to provide clocking to the DDR PLL and PHY portion of the integrated memory controller. 100 MHz.
DDR3_0_REFN	I	DDR	1			VDDQ	DDR3 Clock Reference: Differential-pair input. Used to provide clocking to the DDR PLL and PHY portion of the integrated memory controller. 100 MHz.
DDR3_0_DQECC[7:0]	I/O	DDR	8			VDDQ	DDR3 ECC Bus: Memory Error Correction Code driven along with read and write data.
TOTAL			145				

Table 31-5. DDR1 Signals (Sheet 1 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
DDR3_1_DQ[63:0]	I/O	DDR	64			VDDR	DDR3 Data Bus: Memory read and write data. Data signal interface to the SDRAM data bus. These 64-bit signals have 8-byte lanes, and each byte lane has a corresponding strobe pair.
DDR3_1_MA[15:0]	O	DDR	16			VDDR	DDR3 Memory Address: Provides multiplexed row and column address to memory. Provides the row address for active commands and the column address and Auto Pre-charge bit for read/write commands to select one location out of the memory array in the respective bank. A10 is sampled during a Pre-charge command to determine whether the Pre-charge applies to one bank (A10 LOW) or all banks (A10 HIGH). If only one bank is to be pre-charged, the bank is selected by BA0 - BA2. The address inputs also provide the op-code during MRS or EMRS commands.
DDR3_1_DQS[7:0]	I/O	DDR	8			VDDR	DDR3 Data Strobes: During writes, driven by CDV offset so as to be centered in the data phase. During reads, driven by memory devices edge-aligned with data. The following list matches the data strobe with the data bytes: (DQS_7: DQ[63:56] …. DQS_0: DQ[7:0]). The data strobes may be used in single-ended mode or paired with optional complementary signals DQS_B to provide differential-pair signaling to the system during both reads and writes. A control bit at EMR(1)[A10] enables or disables all complementary data strobe signals.
DDR3_1_DQSECC[0]	I/O	DDR	1			VDDR	DDR3 ECC Strobe: Differential-pair output with read-data ECC, differential-pair input with write-data ECC. Edge-aligned with read-data ECC, centered in write-data ECC.

Table 31-5. DDR1 Signals (Sheet 2 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
DDR3_1_DQSB[7:0]	I/O	DDR	8			VDDR	DDR3 Data Strobes: During writes, driven by CDV offset so as to be centered in the data phase. During reads, driven by memory devices edge-aligned with data. The following list matches the data strobe with the data bytes: (DQS_7: DQ[63:56] …. DQS_0: DQ[7:0]). The data strobes may be used in single-ended mode or paired with optional complementary signals DQS_B to provide differential-pair signaling to the system during both reads and writes. A control bit at EMR(1)[A10] enables or disables all complementary data strobe signals.
DDR3_1_DQSBECC[0]	I/O	DDR	1			VDDR	DDR3 ECC Strobe: Differential-pair output with read-data ECC, differential-pair input with write-data ECC. Edge-aligned with read-data ECC, centered in write-data ECC.
DDR3_1_CK[3:0]	O	DDR	4			VDDR	DDR3 Differential Clock: All address and control input signals are sampled on the crossing of the positive edge of CK and negative edge of CKB. Output (read) data is referenced to the crossings of CK and CKB (both directions of crossing).
DDR3_1_CKB[3:0]	O	DDR	4			VDDR	DDR3 Differential Clock: All address and control input signals are sampled on the crossing of the positive edge of CK and negative edge of CKB. Output (read) data is referenced to the crossings of CK and CKB (both directions of crossing).

Table 31-5. DDR1 Signals (Sheet 3 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
DDR3_1_CKE[3:0]	O	DDR	4			VDDR	DDR3 Clock Enable: (active high). CKE is used for power control of the DRAM devices. For the DRAM Devices: CKE HIGH activates, and CKE LOW deactivates internal clock signals and device input buffers and output drivers. Taking CKE LOW provides Pre-charge Power Down and Self-Refresh operation (all banks idle), or Active Power Down (row Active in any bank). CKE is synchronous for power down entry and exit, and for self-refresh entry. CKE is asynchronous for self-refresh exit. After VREF has become stable during the power-on and initialization sequence, it must be maintained for proper operation of the CKE receiver. For proper self-refresh entry and exit, VREF must be maintained to this input. CKE must be maintained HIGH throughout read and write accesses. Input buffers, excluding CK, CKB, ODT, and CKE are disabled during power down. Input buffers, excluding CKE, are disabled during self-refresh.
DDR3_1_CSB[3:0]	O	DDR	4			VDDR	DDR3 Chip Select: (active low). These signals determine whether a command is valid in a given cycle for the devices connected to it. All commands are masked when CSB is registered HIGH. CSB provides for external rank selection on systems with multiple ranks. CSB is considered part of the command code.

Table 31-5. DDR1 Signals (Sheet 4 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
DDR3_1_ODT[3:0]	O	DDR	4			VDDR	DDR3 On-Die Termination Enable: (active high). ODT (registered HIGH) enables termination resistance internal to the DDR device SDRAM. When the ODT feature is enabled, it is dynamically enabled for the receiver of the data. The SoC does this internally for read data returning from the DRAM devices. For write data to the DRAM devices, the M_ODT[] pins are asserted to enable ODT within the DRAM devices themselves. Because ODT consumes power, when the feature is enabled, it is control dynamically by the SoC. ODT impacts the DQ, DQS, and DM signals. The ODT pin is ignored by the DDR devices if the EMR(1) is programmed to disable ODT. One pin per rank.
DDR3_1_RASB	O	DDR	1			VDDR	DDR3 Row Address Strobe: (active low). Used with CASB and RASB (along with CSB) to define commands. RAS, CAS, and WE (along with CS) define the command being entered.
DDR3_1_CASB	O	DDR	1			VDDR	DDR3 Column Address Strobe: (active low). Used with CAS#, RAS#, and CS# to define commands. RAS, CAS, and WE (along with CS) define the command being entered.
DDR3_1_WEB	O	DDR	1			VDDR	DDR3 Write Enable: (active low). Used with CAS#, RAS#, and CS# to define commands. RAS, CAS, and WE (along with CS) define the command being entered.
DDR3_1_DRAM_PWROK	I	DDR	1			VDDR	DRAM POWER OK. An active high signal indicates that the DDR PHY voltage (VDDR) is good.
DDR3_1_DRAMRSTB	O	DDR	1			VDDR	For resetting the DDR DIMMs.
DDR3_1_VCCA_PWROK	I	DDR	1			VDDR	DDR3 Indication to the DDRIO that the SoC core well voltage is valid. This is connected to the SoC input COREPWROK (except they are different voltages).
DDR3_1_VREF	I	DDR	1		100 1%, PD	VDDR	External Vref from the resistor divider on board.
DDR3_1_ODTPU	I/O	DDR	1			VDDR	DDR3 Compensation Pad. Board trace + External Precision resistor = 275. The resistor is pulled-down to VSS.

Table 31-5. DDR1 Signals (Sheet 5 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
DDR3_1_BS[2:0]	O	DDR	3			VDDR	DDR3 Bank Select: Defines which banks are being addressed within each rank. BA0 - BA2 define to which bank an Active, Read, Write or Pre-charge command is being applied (for 256 MB and 512 MB, BA2 is not applied). The Bank address also determines if the mode register or one of the extended mode registers is to be accessed during an MRS or EMRS command cycle.
DDR3_1_DQPU	I/O	DDR	1			VDDR	DDR3 Compensation Pad. Board trace + External Precision resistor = 35. The resistor is pulled-down to VSS.
DDR3_1_CMDPU	I/O	DDR	1			VDDR	DDR3 Compensation Pad. Board trace + External Precision resistor = 24 or 33.5 depending on target RON. The resistor is pulled-down to VSS.
DDR3_1_MON1P	I/O	DDR	1			VDDR	DDR3 PLL Monitor Port1.
DDR3_1_MON1N	I/O	DDR	1			VDDR	DDR3 PLL Monitor Port1.
DDR3_1_MON2P	I/O	DDR	1			VDDR	DDR3 PLL Monitor Port2.
DDR3_1_MON2N	I/O	DDR	1			VDDR	DDR3 PLL Monitor Port2.
DDR3_1_REFP	I	DDR	1			VDDR	DDRIO MPLL Input Reference Clock. • For SKU 8, this input is connected to VSS on the platform board.
DDR3_1_REFN	I	DDR	1			VDDR	DDRIO MPLL Input Reference Clock. • For SKU 8, this input is connected to VSS on the platform board.
DDR3_1_DQECC[7:0]	I/O	DDR	8			VDDR	DDR3 ECC Bus: Memory Error Correction Code driven along with read and write data.
TOTAL			145				

31.4 Thermal Signals

Table 31-6. Thermal Signals

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
THERMTRIP_N	O, OD	CMOS_V1P0_OD	1		EXT PU	V1P0S	Catastrophic Thermal Trip: When low, this signal indicates that a thermal trip from the processor occurred and has reached an operating temperature that may damage the part.
PROCHOT_B	I/O, OD	CMOS_V1P0_OD	1	2K, PU	EXT PU	V1P0S	Processor Hot: PROCHOT_B goes active when the SoC temperature monitoring sensor(s) detects that the SoC has reached its maximum safe operating temperature. This indicates that the SoC Thermal Control Circuit (TCC) has been activated, if enabled. This signal can also be driven to the processor to activate the TCC. The platform board must ignore this SoC output signal while PMU_PLTRST_B (active-low SoC output) is asserted.
MEMHOT_B	I	CMOS_V1P0	1	2K, PU		V1P0S	Memory Hot: Input from the platform to indicate a memory overheating scenario. The active low signal causes the SoC to perform memory throttling in an attempt to cool the memory. The platform board must ignore this SoC output signal while PMU_PLTRST_B (active-low SoC output) is asserted.
TOTAL			3				

31.5 SVID Signals

Table 31-7. SVID Signals

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
SVID_ALERT_B	I	CMOS_V1P0	1	2K, PU		V1P0S	SVID Alert (Serial Voltage Identification Alert): (active low). Used by VR to signal that the prior request has not reached the requested operating point.
SVID_DATA	I/O, OD	CMOS_V1P0_OD	1	2K, PU		V1P0S	SVID Data (Serial Voltage Identification Data): Bi-Directional signal. Used as data communication interface between the SoC and VR.
SVID_CLK	O, OD	CMOS_V1P0_OD	1	2K, PU		V1P0S	SVID Clock (Serial Voltage Identification Clock): The SoC and VR use this clock for communication on the SVID Data bus. SoC SVID requests are driven out on SVID Data with this clock and are registered in the VR using this for the clock. When the VR responds, it also uses this clock to drive the data.
TOTAL			3				

31.6 Miscellaneous Signals

Table 31-8. Misc. Signals (Sheet 1 of 3)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
NMI/GPIOS_0	I	CMOS_V3P3	1	20K, PD		V3P3S	NMI: This is an NMI event indication to iLB. When operating as NMI event indication pin function (selected via the NMI SMI Event Native GPIO Enable soft strap), the pin is a push-pull. If the NMI interface is not used, the signals can be used as GPIO Port 0.
ERROR2_B/GPIOS_1	O	CMOS_V3P3	1			V3P3S	Root port error collector output. The platform board must ignore this SoC output signal while PMU_PLTRST_B (active-low SoC output) is asserted.If the ERROR2_B interface is not used, the signals can be used as GPIO Port 1.
ERROR1_B/GPIOS_2	O	CMOS_V3P3	1			V3P3S	Root port error collector output. The platform board must ignore this SoC output signal while PMU_PLTRST_B (active-low SoC output) is asserted. If the ERROR1_B interface is not used, the signals can be used as GPIO Port 2.
ERROR0_B/GPIOS_3	O	CMOS_V3P3	1			V3P3S	Root port error collector output. The platform board must ignore this SoC output signal while PMU_PLTRST_B (active-low SoC output) is asserted. If the ERROR0_B interface is not used, the signals can be used as GPIO Port 3.
IERR_B/GPIOS_4	O	CMOS_V3P3	1			V3P3S	Internal Error. Catastrophic error. Requires immediate system shut down. During power-up, IERR_B is valid after the PMU_PLTRST_B (Platform Reset) signal is deasserted by the SoC. Before the Platform Reset is deasserted, the signal may be unstable and falsely signal an internal error. If the IERR_B interface is not used, the signals can be used as GPIO Port 4.

Table 31-8. Misc. Signals (Sheet 2 of 3)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
MCERR_B/GPIOS_5	O	CMOS_V3P3	1			V3P3S	Machine Check Error. Fatal uncorrectable error. During power-up, MCERR_B is valid after the PMU_PLTRST_B (Platform Reset) signal is deasserted by the SoC. Before the Platform Reset is deasserted, the signal may be unstable and falsely signal a machine check error. If the MCERR_B interface is not used, the signals can be used as GPIO Port 5.
UART1_RXD/GPIOS_6	I	CMOS_V3P3	1	20K, PD		V3P3S	UART Port 1 Serial Data Input: Serial data input from the device pin to the receive port for UART port 1. If the UART1_RXD interface is not used, the signals can be used as GPIO Port 6.
UART1_TXD/GPIOS_7	O	CMOS_V3P3	1			V3P3S	UART Port 1 Serial Data Output: Serial data output to the communication peripheral/modem or data set for UART port 1. If the UART1_TXD interface is not used, the signals can be used as GPIO Port 7.
SMB_CLK0/GPIOS_8	I/O, OD	CMOS_V3P3_OD	1	20K, PU	EXT PU	V3P3S	Legacy SMBus Clock - Port 0. External pull-up required. If the SMB_CLK0 interface is not used, the signal can be used as GPIO Port 8.
SMB_DATA0/GPIOS_9	I/O, OD	CMOS_V3P3_OD	1	20K, PU	EXT PU	V3P3S	Legacy SMBus Data - Port 0. External pull-up required. If the SMB_DATA0 interface is not used, the signal can be used as GPIO Port 9.
SMBALRT_N0/GPIOS_10	I/O, OD	CMOS_V3P3_OD	1	20K, PU	EXT PU	V3P3S	Legacy SMBus Alert - Port 0. External pull-up required. If the SMBALRT_N0 interface is not used, the signal can be used as GPIO Port 10.
SMB_DATA1/GPIOS_11	I/O, OD	CMOS_V3P3_OD	1	20K, PU	EXT PU	V3P3S	IOSF SMBus Data - Port 1. External pull-up required. If the SMB_DATA1 interface is not used, the signal can be used as GPIO Port 11.
SMB_CLK1/GPIOS_12/ SPKR	I/O, OD	CMOS_V3P3_OD	1	20K, PU	EXT PU	V3P3S	IOSF SMBus Clock - Port 1. External pull-up required. If the SMB_CLK1 interface is not used, the signal can be used SPKR or can be used as GPIO Port 12.

Table 31-8. Misc. Signals (Sheet 3 of 3)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
SMB_DATA2/GPIOS_13/ UART0_RXD	I/O, OD	CMOS_V3P3_ OD	1	20K, PU	EXT PU	V3P3S	PECI SMBus Data - Port 2. External pull-up required. If the SMB_DATA2 interface is not used, the signal can also be used as UART0_RXD or can be used as GPIO Port 13.
SMB_CLK2/GPIOS_14/ UART0_TXD	I/O, OD	CMOS_V3P3_ OD	1	20K, PU	EXT PU	V3P3S	PECI SMBus Clock - Port 2. External pull-up required. If the SMB_CLK2 interface is not used, the signal can also be used as UART0_TXD or can be used as GPIO Port 14
TOTAL			15				

31.7 SATA2 Signals

Table 31-9. SATA2 Signals (Sheet 1 of 2)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
SATA_GP0/GPIOS_15	I	CMOS_V3P3	1	20K, PU		V3P3S	Serial ATA 0 General Purpose: This is an input pin which can be configured as an interlock switch or as a general purpose I/O depending on the platform. When used as an interlock switch status indication, this signal is driven to 0 to indicate that the switch is closed and to 1 to indicate that the switch is open. If the SATA_GP0 interface is not used, the signals can be used as GPIO Port 15.
SATA_LEDN/GPIOS_16	O, OD	CMOS_V3P3_ OD	1		EXT PU	V3P3S	Serial ATA LED: This is an open-collector output pin driven during SATA command activity. It is to be connected to external circuitry that can provide the current to drive a platform LED. When active, the LED is on. When tri-stated, the LED is off. An external pull-up resistor is required. If the SATA_LEDN interface is not used, the signals can be used as GPIO Port 16.
SATA_TXP[3:0]	O	LV DIFF	4			V1P0S	Serial ATA Ports 3:0 Differential Transmit Pairs: Ports 3:0 3 Gb/s and 1.5 Gb/s.
SATA_TXN[3:0]	O	LV DIFF	4			V1P0S	Serial ATA Ports 3:0 Differential Transmit Pairs: Ports 3:0 3 Gb/s and 1.5 Gb/s.

Table 31-9. SATA2 Signals (Sheet 2 of 2)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
SATA_RXP[3:0]	I	LV DIFF	4			V1P0S	Serial ATA Ports 3:0 Differential Receive Pairs: Ports 3:0 support up 3 Gb/s and 1.5 Gb/s.
SATA_RXN[3:0]	I	LV DIFF	4			V1P0S	Serial ATA Ports 3:0 Differential Receive Pairs: Ports 3:0 support up 3 Gb/s and 1.5 Gb/s.
SATA_REFCLKP	I	LV DIFF	1			V1P0S	Serial ATA 100 MHz Differential Clock: Reference clock 100 MHz differential signal from a clock chip. If unused, tie to ground through a 10 kΩ resistor.
SATA_REFCLKN	I	LV DIFF	1			V1P0S	Serial ATA 100 MHz Differential Clock: Reference clock 100 MHz differential signal from a clock chip. If unused, tie to ground through a 10 kΩ resistor.
SATA_OBSP SATA_OBSN	O	Analog	2			V1P0S	SATA RCOMP: Connect the SATA_OBSP pin to the SATA_OBSN pin using a 402-Ω ±1% resistor.
TOTAL			22				

31.8 SATA3 Signals

Table 31-10. SATA3 Signals (Sheet 1 of 2)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
SATA3_GP0/GPIOS_17	I	CMOS_V3P3	1	20K, PU		V3P3S	Serial ATA 0 General Purpose: This is an input pin which can be configured as an interlock switch or as a general purpose I/O depending on the platform. When used as an interlock switch status indication, this signal is driven to 0 to indicate that the switch is closed and to 1 to indicate that the switch is open. If the SATA3_GP0 interface is not used, the signals can be used as GPIO Port 17.
SATA3_LEDN/GPIOS_18	O, OD	CMOS_V3P3_OD	1		EXT PU	V3P3S	Serial ATA LED: This is an open-collector output pin driven during SATA command activity. It is to be connected to external circuitry that can provide the current to drive a platform LED. When active, the LED is on. When tri-stated, the LED is off. An external pull-up resistor is required. If SATA3_LEDN interface is not used, the signals can be used as GPIO Port 18.
SATA3_TXP[1:0]	O	LV DIFF	2			V1P0S	Serial ATA Ports 1:0 Differential Transmit Pairs: Ports 1:0 support up to 6 Gb/s.
SATA3_TXN[1:0]	O	LV DIFF	2			V1P0S	Serial ATA Ports 1:0 Differential Transmit Pairs: Ports 1:0 support up to 6 Gb/s.
SATA3_RXP[1:0]	I	LV DIFF	2			V1P0S	Serial ATA Ports 1:0 Differential Receive Pairs: Ports 1:0 support up to 6 Gb/s.
SATA3_RXN[1:0]	I	LV DIFF	2			V1P0S	Serial ATA Ports 1:0 Differential Receive Pairs: Ports 1:0 support up to 6 Gb/s.
SATA3_REFCLKP	I	LV DIFF	1			V1P0S	Serial ATA 100 MHz Differential Clock: Reference clock 100 MHz differential signal from a clock chip. If unused, tie to ground through a 10 kΩ resistor.

Table 31-10. SATA3 Signals (Sheet 2 of 2)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
SATA3_REFCLKN	I	LV DIFF	1			V1P0S	Serial ATA 100 MHz Differential Clock: Reference clock 100 MHz differential signal from a clock chip. If unused, tie to ground through a 10 kΩ resistor.
SATA3_OBSP SATA3_OBSN	O	Analog	2			V1P0S	SATA3 RCOMP: Connect the SATA3_OBSP pin to the SATA3_OBSN pin using a 402-Ω ±1% resistor.
TOTAL			14				

31.9 PCIe Signals

Table 31-11. PCIe Signals

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
FLEX_CLK_SE0/ GPIOS_19	O	CMOS_V3P3	1			V3P3S	Single-ended, flexible, general-purpose, 25-MHz clock output. Can be programmed to be 33 MHz or disabled using the CCU Dividers Control Register (DIV_CTRL) located at sideband register Port 40h, offset 0Ch. If the FLEX_CLK_SE0 interface is not used, the signals can be used as GPIO Port 19.
FLEX_CLK_SE1/ GPIOS_20	O	CMOS_V3P3	1			V3P3S	Single-ended, flexible, general-purpose, 25-MHz clock output. Can be programmed to be 33 MHz or disabled using the CCU Dividers Control Register (DIV_CTRL) located at sideband register Port 40h, offset 0Ch. If the FLEX_CLK_SE1 interface is not used, the signals can be used as GPIO Port 20.
PCIE_TXP[15:0]	O	LV DIFF	16			V1P0S	PCI Express* Transmit: Differential-pair output. 2.5GT/s and 5.0GT/s data rates supported.
PCIE_TXN[15:0]	O	LV DIFF	16			V1P0S	PCI Express Transmit: Differential-pair output. 2.5GT/s and 5.0GT/s data rates supported.
PCIE_RXP[15:0]	I	LV DIFF	16			V1P0S	PCI Express Receive: Differential-pair input. 2.5GT/s and 5.0GT/s data rates supported.
PCIE_RXN[15:0]	I	LV DIFF	16			V1P0S	PCI Express Receive: Differential-pair input. 2.5GT/s and 5.0GT/s data rates supported.
PCIE_REFCLKN	I	LV DIFF	1			V1P0S	PCI Express Reference Clock: Differential-pair input 100 MHz. PCIe* PLL Differential reference clock for PCIe PLL.
PCIE_REFCLKP	I	LV DIFF	1			V1P0S	PCI Express Reference Clock: Differential-pair input 100 MHz. PCIe PLL Differential reference clock for PCIe PLL.
PCIE_OBSP PCIE_OBSN	O	Analog	2			V1P0S	PCIE RCOMP: Connect the PCIE_OBSP pin to the PCIE_OBSN pin using a 402-Ω ±1% resistor.
TOTAL			70				

31.10 GbE, SMBus, and NC-SI Signals

Table 31-12. GbE, SMBus, and NC-SI Signals (Sheet 1 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
GBE_TXP[3:0]	O	LV DIFF	4			V1P0A	SerDes/SGMII Serial Data output Port: Differential SGMII/SerDes Transmit interface.
GBE_TXN[3:0]	O	LV DIFF	4			V1P0A	SerDes/SGMII Serial Data output Port: Differential SGMII/SerDes Transmit interface.
GBE_RXP[3:0]	I	LV DIFF	4			V1P0A	SerDes/SGMII Serial Data input Port: Differential SGMII/ SerDes Receive interface.
GBE_RXN[3:0]	I	LV DIFF	4			V1P0A	SerDes/SGMII Serial Data input Port: Differential SGMII/ SerDes Receive interface.
GBE_REFCLKP GBE_REFCLKN	I	LV DIFF	1			V1P0A	GbE 100 MHz differential clock with 100 ppm maximum jitter. External SerDes/SGMII differential 100 MHz reference clock from an external generator. This clock must be powered from the Suspend (SUS) power well. When the device is enabled for 2.5-GbE operation, the standard 100-MHz reference clock must be replaced with a 125-MHz reference clock.
GBE_OBSP GBE_OBSN	O	Analog	2			V1P0A	GBE RCOMP: Connect the GBE_OBSP pin to the GBE_OBSN pin using a 402-Ω ±1% resistor.
GBE_SMBD/ NCSI_TX_EN	I/O, OD	CMOS_V3P3_ OD	1	20K, PU	EXT PU	V3P3A	GbE SMBus Clock. One clock pulse is generated for each data bit transferred. An external pull-up resistor required. Resistor value is calculated based on the bus load. (Refer to the Platform Design Guide). If the GBE_SMBD interface is not used, the signals can be used as NCSI_TX_EN Transmit Enable (input). ***Note:*** If not used, have an external pull-down resistor.

Table 31-12. GbE, SMBus, and NC-SI Signals (Sheet 2 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
GBE_SMBCLK/ NCSI_CLK_IN	I/O, OD	CMOS_V3P3_ OD	1	20K, PU	EXT PU	V3P3A	GbE SMBus Clock. One clock pulse is generated for each data bit transferred. An external pull-up resistor required. If the GBE_SMBCLK interface is not used, the signals can be used as the NCSI_CLK_IN signal. As an input signal, the NCSI_CLK_IN must be connected to the 50-MHz NC-SI REF_CLK generator on the platform board. This same signal pin can be programmed to provide the 50-MHz NC-SI REF_CLK for the NC-SI devices on the platform board including the SoC. If so programmed, the NCSI_CLK_IN pin also functions as the "NCSI_CLK_OUT" of the SoC. ***Note:*** If not used, have an external pull-down resistor. Also, this clock is in addition to and separate from the XTAL clock.
GBE_SMBALRT_N/ NCSI_CRS_DV	I/O, OD	CMOS_V3P3_ OD	1	20K, PU	EXT PU	V3P3A	GbE SMBus Alert. Acts as an interrupt of a slave device on SMBus. External pull-up resistor required. If the GBE_SMBALRT_N interface is not used, the signals can be used as NCSI_CRS_DV Carrier Sense/Receive Data Valid (CRS/DV).
GBE_SDP0_0/ GPIO_SUS17	I/O	CMOS_V3P3	1	20K, PU		V3P3A	GbE Port 0 SW Defined Pin 0: The SDP pins are reserved pins that are software programmable write/read input/output capability. These default to inputs upon power-up, but may have their direction and output values defined in the EEPROM. The SDP bits may be mapped to the General Purpose Interrupt bits when configured as inputs. The SDP0_0 pin can be used as a watchdog output indication. If the GBE_SDP0_0 interface is not used, the signal can be used as GPIO SUS Port 17.

Table 31-12. GbE, SMBus, and NC-SI Signals (Sheet 3 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
GBE_SDP0_1/ GPIO_SUS18/ NCSI_ARB_IN	I/O	CMOS_V3P3	1	20K, PU		V3P3A	GbE Port 0 SW Defined Pin1: The SDP pins are reserved pins that are software programmable write/read input/output capability. These default to inputs upon power-up, but may have their direction and output values defined in the EEPROM. The SDP bits may be mapped to the General Purpose Interrupt bits when configured as inputs. The SDP0_1 pin can be used as a watchdog output indication. The SDP0_1 pin can be used as a strapping option to disable PCIe* Function 0. In this case it is latched at the rising edge of PE_RST# or In-Band PCIe Reset. If GBE_SDP0_1 the interface is not used, the signal can be used as GPIO SUS Port 18. If none of the above functions are used, the signal can be used as NCSI_ARB_IN Arbitration Input.
GBE_LED0/GPIO_SUS19	O	CMOS_V3P3	1			V3P3A	GBE_LED[3:0]Programming: 0000: Port0linkup 0001: Port1linkup 0010: Port2linkup 0011: Port3linkup 0100: Port0activity 0101: Port1activity 0110: Port2activity 0111: Port3activity 1000: Ports0-3“linkup” 1001: Ports0-1“linkup” 1010: Ports0-3activity 1011: Ports0-1activity If the GBE_LED[3:0] interface is not used, the signals can be used as GPIO SUS Port [22:19].
GBE_LED1/GPIO_SUS20	O	CMOS_V3P3	1			V3P3A	
GBE_LED2/GPIO_SUS21	O	CMOS_V3P3	1			V3P3A	
GBE_LED3/GPIO_SUS22	O	CMOS_V3P3	1			V3P3A	
NCSI_RXD1/ GPIO_SUS23	O	CMOS_V3P3	1			V3P3A	NC-SI Receive Data 1. Data signal to the Manageability Controller (MC). ***Note:*** NCSI_RXD1 is also a sampled pin strap that defines whether or not the GBE needs power when the system is in S5 state. Settings in the EEPROM will either enable or disable the WOL feature. Different than previous generations of WOL implementations, the driver has no control of this behavior. Refer to Section 16.2, “Pin-Based (Hard) Straps” on page 357.

Table 31-12. GbE, SMBus, and NC-SI Signals (Sheet 4 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
GBE_MDIO0_I2C_CLK/ GPIO_SUS24	I/O, OD	CMOS_V3P3_ OD	1	20K, PU	EXT PU	V3P3A	Gigabit Ethernet Controller Management Channel 0 Clock (out): Serial clock for the management channel. Can also be configured to SFP/I2C (OD) Clock. If the GBE_MDIO0_I2C_CLK interface is not used, the signal can be used as GPIO SUS Port 24.
GBE_MDIO0_I2C_DATA/ GPIO_SUS25	I/O, OD	CMOS_V3P3_ OD	1	20K, PU	EXT PU	V3P3A	Gigabit Ethernet Controller Management Channel 0 Data (T/S): Serial data for the management channel. Can also be configure to SFP/I2C (OD) data. If the GBE_MDIO1_I2C_DATA interface is not used, the signal can be used as GPIO SUS Port 25.
GBE_MDIO1_I2C_CLK/ GPIO_SUS26/ NCSI_TXD1	I/O, OD	CMOS_V3P3_ OD	1	20K, PU	EXT PU	V3P3A	Gigabit Ethernet Controller Management Channel 1 Clock (out): Serial clock for the management channel. Can also be configure to SFP/I2C (OD) Clock. If the GBE_MDIO1_I2C_CLK interface is not used, the signal can be used as GPIO SUS Port 26. If none of the above functions are used, the signal can be used as NCSI_TXD1 Transmit Data 1. Data signals from the MC. ***Note:*** If not used, have an external pull-up resistor.
GBE_MDIO1_I2C_DATA/ GPIO_SUS27/ NCSI_TXD0	I/O, OD	CMOS_V3P3_ OD	1	20K, PU	EXT PU	V3P3A	Gigabit Ethernet Controller Management Channel 1 Data (T/S): Serial data for the management channel. Can also be configured to SFP/I2C (OD) data. If the GBE_MDIO1_I2C_DATA interface is not used, the signal can be used as GPIO SUS Port 27. If none of the above functions are used, the signal can be used as NCSI_TXD0 Transmit Data 0. Data signals from the MC. ***Note:*** If not used, have an external pull-up resistor.

Table 31-12. GbE, SMBus, and NC-SI Signals (Sheet 5 of 5)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
GPIO_SUS1/NCSI_RXD0	I/O	CMOS_V3P3	1	20K, PU		V3P3A	SUS Well GPIO_1: General purpose Customer I/O. If GPIO_SUS1 is not used, the signal can be used as NCSI_RXD0 Receive Data 0 signal to the Manageability Controller (MC). This pin is also a pin-strap input. If sensed low, the 2.5-GbE capability, if available, is disabled. This pin must be sampled high for the 2.5-GbE capability to function. This pin is temporarily pulled-down internally during the sample period. An external pull-up resistor is needed during the sample period to enable 2.5 GbE. Refer to Section 16.2, "Pin-Based (Hard) Straps" on page 357.
STRAP_NCSI_EN/ Y59_RSVD/ NCSI_ARB_OUT	O	CMOS_V3P3	1			V3P3A	NC-SI hardware arbitration token output pin. ***Note:*** This pin is also a hard pin strap. When it is a logic high at power-up, it indicates the NC-SI interface is to be used rather than the GBE_SMBus. Refer to Section 16.2, "Pin-Based (Hard) Straps" on page 357.
TOTAL			35				

Table 31-13. GbE EEPROM Signals

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
GBE_EE_DI/ GPIO_SUS13	O	CMOS_V3P3	1			V3P3A	GbE EEPROM Data Input: Data is output to EEPROM. If the GBE_EE_DI interface is not used, the signal can be used as GPIO SUS Port 13.
GBE_EE_DO/ GPIO_SUS14	I	CMOS_V3P3	1	20K, PU		V3P3A	GbE EEPROM Data Output: Data is input from EEPROM. If the GBE_EE_DO interface is not used, the signal can be used as GPIO SUS Port 14.
GBE_EE_SK/ GPIO_SUS15	O	CMOS_V3P3	1			V3P3A	GbE EEPROM Serial Clock: Serial clock output to EEPROM Operates at ~2 MHz. If the GBE_EE_SK interface is not used, the signal can be used as GPIO SUS Port 15.
GBE_EE_CS_N/ GPIO_SUS16	O	CMOS_V3P3	1			V3P3A	GbE EEPROM Chip Select: Chip select Output to EEPROM. If the GBE_EE_CS_N interface is not used, the signal can be used as GPIO SUS Port 16.
TOTAL			4				

31.11 LPC Interface Signals

Table 31-14. LPC Signals (Sheet 1 of 2)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
LPC_AD0	I/O	CMOS_V3P3	1	20K, PU		V3P3S	LPC Address/Data: Multiplexed Command, Address, Data.
LPC_AD1	I/O	CMOS_V3P3	1	20K, PU		V3P3S	LPC Address/Data: Multiplexed Command, Address, Data.
LPC_AD2	I/O	CMOS_V3P3	1	20K, PU		V3P3S	LPC Address/Data: Multiplexed Command, Address, Data.
LPC_AD3	I/O	CMOS_V3P3	1	20K, PU		V3P3S	LPC Address/Data: Multiplexed Command, Address, Data.
LPC_FRAMEB	O	CMOS_V3P3	1			V3P3S	LPC Frame: (active low). Output signal that indicates the start of an LPC cycle or an abort. ***Note:*** The LPC controller does not implement DMA or bus mastering cycles.
LPC_CLKOUT0	O	CMOS_V3P3	1			V3P3S	LPC Clock: These signals are the clocks driven by the processor to the LPC devices. Each clock can support up to two loads. ***Note:*** If the primary boot device is connected via the LPC interface, it should use LPC_CLKOUT[0]. Using the LPC interface for the boot device is not supported at this time and may not ever be supported by this Intel product. Only use the SPI interface for boot device connection.
LPC_CLKOUT1	O	CMOS_V3P3	1			V3P3S	LPC Clock: These signals are the clocks driven by the processor to the LPC devices. Each clock can support up to two loads. ***Note:*** If the primary boot device is connected via the LPC interface, it uses LPC_CLKOUT[0]. Using the LPC interface for the boot device is not supported at this time and may not ever be supported by this Intel product. Only use the SPI interface for boot device connection.

Table 31-14. LPC Signals (Sheet 2 of 2)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
LPC_CLKRUNB	I/O, OD	CMOS_V3P3_OD	1		EXT PU	V3P3S	Clock Run: (active low). Bi-directional signal that gates the operation of the LPC_CLKOUTx. Once an interrupt sequence has started, LPC_CLKRUN_B remains asserted to allow the LPC_CLKOUTx to run.
ILB_SERIRQ/GPIOS_29	I/O	CMOS_V3P3	1	20K, PU		V3P3S	Serial Interrupt Request: This pin conveys the serial interrupt protocol. If the ILB_SERIRQ interface is not used, the signals can be used as GPIO Port 29.
TOTAL			9				

31.12 RTC Well Signals

Table 31-15. RTC Well Signals (Sheet 1 of 2)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
RTEST_B	I	CMOS_V3P3	1		EXT RC Circuit	VRTC3P0	RTC Battery Test: Active-low signal. An external RC circuit creates a time delay for the signal such that it goes high sometime after the battery voltage is valid. The RC time delay must be in the 10-20 ms range. This allows the SoC to detect when a new battery has been installed. This signal is internally asserted after the suspend power is up if the coin cell battery is weak. When active, this signal also resets some bits in the RTC well that are otherwise not reset by PLTRST_B, or SRTCRST_B. ***Note:*** This signal may also be used for debug purposes, as part of an XDP port. Unless entering a test mode, the RTEST_B input must always be high when all other non-RTC power planes are on. This signal is in the RTC power well. The time delay parameters are provided in Chapter 7, "SoC Reset and Power Supply Sequences."
RSMRST_B	I	CMOS_V3P3	1		EXT PU	VRTC3P0	Resume Well Reset: (active low). Input asserted by the External Circuitry (EC) to reset the registers and components in the SUS power well. An external RC circuit is required to ensure that the SUS power well voltage is valid before the deassertion of the RSMRST_B signal.
COREPWROK	I	CMOS_V3P3	1			VRTC3P0	Core Power OK. Input asserted by the External Circuitry (EC) to indicate on that the power supplied to the core is stable. PWROK can be driven asynchronously. The EC typically uses PWROK to produce the PERST_B signal on the PCI Express* interfaces. The power associated with the PCI Express circuitry needs to be valid for at least 99 ms before COREPWROK assertion to comply with the PCI Express 100-ms requirement for system reset deassertion.

Table 31-15. RTC Well Signals (Sheet 2 of 2)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
SRTCRST_B	I	CMOS_V3P3	1		EXT RC Circuit	VRTC3P0	RTC Well Secondary Reset: (active low). Normally held high. Can be driven low with external circuitry to test the RTC power well and reset certain register bits in the RTC power well registers that are not reset by the Resume Well Reset signal RSMRST_B.
BRTCX1_PAD	I/O	Analog	1			VRTC3P0	RTC Crystal Input Pad 1: Pad 1 connection for the RTC external 32.768 kHz crystal and associated circuitry.
BRTCX2_PAD	I/O	Analog	1			VRTC3P0	RTC Crystal Input Pad 2. Pad 2 connection for the RTC external 32.768 kHz crystal and associated circuitry.
BVCCRTC_EXTPAD	I/O	Analog	1		EXT 0.1 µF Capacitor to VSS	VRTC3P0	RTC Internal Voltage Regulator External Pad: Requires 0.1 µF capacitor connected to VSS on the platform board to de-couple the RTC internal voltage regulator.
TOTAL			8				

31.13 GPIO SUS Signals

Table 31-16. GPIO SUS Signals

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
GPIO_SUS0	I/O	CMOS_V3P3	1	20K, PU		V3P3A	SUS Well GPIO_0: General purpose legacy I/O.
GPIO_SUS2	I/O	CMOS_V3P3	1	20K, PU		V3P3A	SUS Well GPIO_2: General purpose legacy I/O.
TOTAL			2				

31.14 PMU Signals

Table 31-17. PMU Signals (Sheet 1 of 3)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
CPU_RESET_B/ GPIO_SUS3	O	CMOS_V3P3	1			V3P3A	CPU Reset: Combined CPU reset for ITP debugger. This is the logical AND of all core reset signals. If the CPU_RESET_B interface is not used, the signal can be used as GPIO SUS Port 3.
SUSPWRDNACK/ GPIO_SUS4	O	CMOS_V3P3	1			V3P3A	Active high. Asserted by the SoC on behalf of the PMC when it does not require the SoC suspend well to be powered. This pin requires a pull-up to VccSUS. If the SUSPWRDNACK interface is not used, the signal can be used as GPIO SUS Port 4.
PMU_SUSCLK/ GPIO_SUS5	O	CMOS_V3P3	1			V3P3A	Output of the RTC generator circuit (32.768 kHz). SUSCLK has a duty cycle that can be as low as 30% or as high as 70%. If the PMU_SUSCLK interface is not used, the signal can be used as GPIO SUS Port 5.
PMU_SLP_DDRVTT_B/ GPIO_SUS6	O	CMOS_V3P3	1			V3P3A	Controls the power of DRAM. If the PMU_SLP_DDRVTT_B interface is not used, the signal can be used as GPIO SUS Port 6.
PMU_SLP_S45_B	O	CMOS_V3P3	1			V3P3A	Power plane control. Shuts power to non-critical systems when in the S5 (Soft-Off) state.
PMU_SLP_S3_B	O	CMOS_V3P3	1			V3P3A	Power plane control. Shuts power to non-critical systems when in the S3 (Suspend To RAM) state.
PMU_SLP_LAN_B/ GPIO_SUS7	O	CMOS_V3P3	1			V3P3A	LAN Subsystem Sleep Control: This active-low output signal is non-functional. It is always high indicating that the PHY device must be powered. If this signal is not needed for the platform board design, it can be re-configured to function as GPIO_SUS7.

Table 31-17. PMU Signals (Sheet 2 of 3)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
PMU_WAKE_B/ GPIO_SUS8	I	CMOS_V3P3	1	20K, PU		V3P3A	PCI Express* Wake-up Event: (active low). Open-Drain input signal that is asserted by a PCI Express port indicating it wants to wake-up the system. This is a single signal, named WAKE# by the PCI Express specification that can be driven by any of the PCI Express devices implemented on the platform board. The device indicating the wake-up drives this signal low. If the PMU_WAKE_B interface is not used, the signal can be used as GPIO SUS Port 8.
PMU_PWRBTN_B/ GPIO_SUS9	I	CMOS_V3P3	1	20K, PU		V3P3A	Causes SMI# or SCI to indicate to the system request to go to a sleep state. If the system is in the S5 (Soft-Off) state, it causes a wake event. If PWRBTNB is pressed for more than 4 seconds, it causes an unconditional transition (power button override) to the S5 state. If the PMU_PWRBTN_B interface is not used, the signal can be used as GPIO SUS Port 9.
PMU_RESETBUTTON_B/ GPIOS_30	I	CMOS_V3P3	1	20K, PU		V3P3S	By default • PMU_RESETBUTTON_B is a debounced edged signal to the power managment controller. • This signal tells the power management controller to perform a reset. Once microcode patch 11F (or later) is applied • PMU_RESETBUTTON_B becomes a debounced level signal to the power managment controller. • This signal tells the power management controller to perform (just like in the default mode) but hold the IA cores in reset until PMU_RESETBUTTON_B is deasserted.

Table 31-17. PMU Signals (Sheet 3 of 3)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
PMU_PLTRST_B	O	CMOS_V3P3	1			V3P3A	Platform Reset; The SoC asserts PLTRST_B as the main SoC platform reset.
SUS_STAT_B/ GPIO_SUS10	O	CMOS_V3P3	1			V3P3A	This signal is asserted by the SoC to indicate that the system is entering a low-power state soon. This can be monitored by devices with memory that need to switch from normal refresh to suspend refresh mode. It can also be used by other peripherals as an indication that the devices should isolate the outputs that may be going to powered-off planes. If the SUS_STAT_B interface is not used, the signal can be used as GPIO SUS Port 10.
TOTAL			12				

31.15 USB 2 Signals

Table 31-18. USB 2 Signals (Sheet 1 of 2)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
USB_DP[3:0]	I/O	LV DIFF	4			V1P0A	Universal Serial Bus Port [3:0] Differential: These differential pairs are used to transmit Data/Address/Command signals for ports 0, 1, 2, and 3.
USB_DN[3:0]	I/O	LV DIFF	4			V1P0A	Universal Serial Bus Port [3:0] Differential: These differential pairs are used to transmit Data/Address/Command signals for ports 0, 1, 2, and 3.
USB_REFCLKN	I	LV DIFF	1			V1P0A	USB Clock 96 MHz. Differential reference input clock from an external clock chip.
USB_REFCLKP	I	LV DIFF	1			V1P0A	USB Clock 96 MHz. Differential reference input clock from an external clock chip.
USB_OC0_B/ GPIO_SUS11	I	CMOS_V3P3	1	20K, PU		V3P3A	Over Current Indicator: This signal sets the corresponding bit in the USB controller to indicate that an over-current condition has occurred. OC0 covers ports 0-3. These signals are NOT 5V tolerant. If the USB_OC0_B interface is not used, the signal can be used as GPIO SUS Port 11.

Table 31-18. USB 2 Signals (Sheet 2 of 2)

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
USB_RCOMPO	O	Analog	1		EXT PD	V1P0A	USB Resistor Bias. Analog connection point for an external resistor. Short the USB_RCOMPO and the USB_RCOMPI pins together and connect to a 44.745-Ω ±1% resistor to ground.
USB_RCOMPI	I	Analog	1		EXT PD	V1P0A	USB Resistor Bias Complement. Analog connection point for an external resistor. See the description for USB_RCOMPO.
USB_OBSP	O	Analog	1			V1P0A	Reserved for Intel. Make no board connection to this pin.
VSSA_USB	NA	NA	2			NA	The platform board must connect to VSS.
TOTAL			16				

31.16 SPI Signals

Table 31-19. SPI Signals

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
SPI_CS0_B	O	CMOS_V3P3	1			V3P3A	This signal is the SPI Chip Select 0 output.
SPI_CS1_B/ GPIO_SUS12	O	CMOS_V3P3	1			V3P3A	This signal is the SPI Chip Select 1 output. If the SPI_CS1_B interface is not used, the signal can be used as GPIO SUS Port 15.
SPI_MISO	I	CMOS_V3P3	1	20K, PU		V3P3A	Data input from the SPI to the SoC. This signal has a weak internal pull-up that is always ON except during leakage test mode.
SPI_MOSI	I/O	CMOS_V3P3	1	20K, PD		V3P3A	Data output from the SoC to the SPI.
SPI_CLK	O	CMOS_V3P3	1			V3P3A	SPI clock signal. The default is 20 MHz, but can be set to 33 MHz. During bus idle, the SoC drives the clock signal low.
TOTAL			5				

31.17 GPIO DFX Signals

Table 31-20. GPIO DFX Signals

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
DFX_PORT_CLK0	I/O	CMOS_V1P0	1			V1P0S	DFx Interface Clock 0: Design-for-debug (also test, manufacture, and validation) clock.
DFX_PORT_CLK1	I/O	CMOS_V1P0	1			V1P0S	DFx Interface Clock 1: Design-for-debug (also test, manufacture, and validation) clock
DFX_PORT[15:0]	I/O	CMOS_V1P0	16			V1P0S	DFx Interface I/O Data.
CTBTRIGINOUT	I/O	CMOS_V1P0	1	20K, PU		V1P0S	I/O pin trigger for the logic analyzer. If unused, leave as NC.
CTBTRIGOUT	O	CMOS_V1P0	1			V1P0S	Output pin trigger for the logic analyzer. If unused, leave as NC.
RCOMP_CORE_LVT	N/A	Analog	1			V1P0S	Resistor Bias Complement. Analog connection point for an external resistor. Used to set transmit currents and internal load resistors.
TOTAL			21				

31.18 Clock Receiver Signals

Table 31-21. Clock Receiver Signals

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
HPLL_REFP	I	LV DIFF	1			V1P0S	Host Clock Reference: Differential-pair input. Used to provide clocking to the processor core, integrated memory controller, and most of the integrated legacy devices. 100 MHz.
HPLL_REFN	I	LV DIFF	1			V1P0S	Host Clock Reference: Differential-pair input. Used to provide clocking to the processor core, integrated memory controller, and most of the integrated legacy devices. 100 MHz.
CLK14_IN	I	CMOS_V3P3	1	20K, PU		V3P3S	14.31838 MHz input clock.
TOTAL			3				

31.19 Tap Port/ITP Signals

Table 31-22. Pins with Shared Functions

Signal Name	I/O Type	I/O Buffer Type	Ball Count	Internal Resistor PU/PD	External Resistor PU/PD	Power Rail	Description
TCK	I	CMOS_V1P0	1	2K, PD	51, EXT PD	V1P0A	JTAG Test Clock for the JTAG controller.
TRST_B	I	CMOS_V1P0	1	2K, PU		V1P0A	JTAG Reset. Resets the JTAG controller when asserted. The signal has an internal pull-up resistor to comply with 1149.1. An external 51-Ω 1% pull-down resistor is required to disable JTAG and keep TAP in safe mode.
TMS	I	CMOS_V1P0	1	2K, PU	51, EXT PD	V1P0A	JTAG Test Mode Select. Selects the state of the JTAG controller. Sampled with the rising edge of JTCK.
TDI	I	CMOS_V1P0	1	2K, PU		V1P0A	JTAG Test Data In. Sampled with the rising edge of JTCK.
TDO	O	CMOS_V1P0	1		51, EXT PU	V1P0A	JTAG Test Data Out.
CX_PRDY_B	O, OD	CMOS_V1P0	1	2K, PU		V1P0S	Probe Mode Ready: CPU response to XXPREQ_B assertion. Indicates CPU is in probe mode. Input unused.
CX_PREQ_B	I	CMOS_V1P0	1	2K, PU		V1P0S	Probe Mode Request: Assertion is a request for the CPU to enter probe mode. The CPU responds with XXPRDY_B assertion once it has entered. The XXPREQ_B can be enabled to cause the CPU to break from C6. External 56 Ω resistor to Vccp.
TOTAL			7				

31.20 Reserved Signals

Table 31-23. Reserved Signals

Signal Name	Ball Count	External Resistor PU/PD	Power Rail	Description
AA47_RSVD	1	NC		Reserved Signal
AB63_RSVD	1	NC		Reserved Signal
AC26_RSVD	1	NC		Reserved Signal
AC25_RSVD	1	NC		Reserved Signal
AD53_RSVD	1	10K Ω PU	VRTC3P0	Reserved Signal
AP21_RSVD	1	NC		Reserved Signal
AP20_RSVD	1	NC		Reserved Signal
AL34_RSVD	1	NC		Reserved Signal
AJ34_RSVD	1	NC		Reserved Signal
AG60_RSVD	1	0 Ω PD		Reserved Signal
AR51_RSVD	1	NC		Reserved Signal
AR53_RSVD	1	NC		Reserved Signal
AR54_RSVD	1	NC		Reserved Signal
L38_RSVD	1	NC		Reserved Signal
J38_RSVD	1	NC		Reserved Signal
AU34_RSVD	1	NC		Reserved Signal
AT34_RSVD	1	NC		Reserved Signal
Y59_RSVD/NCSI_ARB_OUT	1	See Description		Reserved Signal Signal can be used as NCSI_ARB_OUT Arbitration Output. Also acts as pin strap at suspwrgood to indicate whether this is SMBus mode (A0) or NC-SI mode. Pull-up to enable NC-SI. The SoC internally pulls this pin down to maintain A0 compatibility (pin was formerly an unused output).
R37_RSVD	1	NC		Reserved Signal
P38_RSVD	1	NC		Reserved Signal
AT51_RSVD	1	NC		Reserved Signal
TOTAL	21			

Note: NC indicates that no connection is to be made to this signal pin on the platform board.

31.21 Signal Pins with Shared Functions or GPIO

The following lists of signal pins/balls have shared functions. The shared functions can be in the form of a hard-strap pin that is sampled at reset time, (see Table 16-1, "Hard Pin Straps" on page 357), one or two normal functions that need some kind of attention to configure, or a Customer General-Purpose I/O (GPIO) signal if the normal function(s) is not needed.

Table 31-24 is the list for signal pins in the SoC Core Power Well, and Table 31-25 on page 614 is the list for signal pins in the SoC SUS Power Well.

Table 31-24. Signal Pins with Shared Functions - Core Power Well (Sheet 1 of 2)

SoC Signal Pin Name	Functional Options and Occurrences	SoC Signal Represented by Pin	Direction (I/O)	Internal Pull-up (PU) or Pull-down (PD)	Power Rail
NMI	As BIOS Starts	NMI	I	20K PD	V3P3S
	SC_USE_SEL = 1	GPIOS_0	Set by the Software (SW)	Design Specific	V3P3S
ERROR2_B	Strap Sampling	Strap: Reserved for Intel	I	20K PD	V3P3S
	As BIOS Starts	ERROR2_B	O	None	V3P3S
	SC_USE_SEL = 1	GPIOS_1	Set by SW	Design Specific	V3P3S
ERROR1_B	Strap Sampling	Strap: Reserved for Intel	I	20K PD	V3P3S
	As BIOS Starts	ERROR1_B	O	None	V3P3S
	SC_USE_SEL = 1	GPIOS_2	Set by SW	Design Specific	V3P3S
ERROR0_B	Strap Sampling	Strap: Reserved for Intel	I	20K PD	V3P3S
	As BIOS Starts	ERROR0_B	O	None	V3P3S
	SC_USE_SEL = 1	GPIOS_3	Set by SW	Design Specific	V3P3S
IERR_B	As BIOS Starts	IERR_B	O	None	V3P3S
	SC_USE_SEL = 1	GPIOS_4	Set by SW	Design Specific	V3P3S
MCERR_B	As BIOS Starts	MCERR_B	O	None	V3P3S
	SC_USE_SEL = 1	GPIOS_5	Set by SW	Design Specific	V3P3S
UART1_RXD	As BIOS Starts	UART1_RXD	I	20K PD	V3P3S
	SC_USE_SEL = 1	GPIOS_6	Set by SW	Design Specific	V3P3S
UART1_TXD	Strap Sampling	0 = Override SPI Flash Descriptor Security	I	20K PU	V3P3S
	As BIOS Starts	UART1_TXD	O	None	V3P3S
	SC_USE_SEL = 1	GPIOS_7	Set by SW	Design Specific	V3P3S
SMB_CLK0	As BIOS Starts	SMB_CLK0	I/O, OD	20K PU	V3P3S
	SC_USE_SEL = 1	GPIOS_8	Set by SW	Design Specific	V3P3S
SMB_DATA0	As BIOS Starts	SMB_DATA0	I/O, OD	20K PU	V3P3S
	SC_USE_SEL = 1	GPIOS_9	Set by SW	Design Specific	V3P3S
SMBALRT_N0	As BIOS Starts	SMBALRT_N0	I/O, OD	20K PU	V3P3S
	SC_USE_SEL = 1	GPIOS_10	Set by SW	Design Specific	V3P3S
SMB_DATA1	As BIOS Starts	SMB_DATA1	I/O, OD	20K PU	V3P3S
	SC_USE_SEL = 1	GPIOS_11	Set by SW	Design Specific	V3P3S

Table 31-24. Signal Pins with Shared Functions - Core Power Well (Sheet 2 of 2)

SoC Signal Pin Name	Functional Options and Occurrences	SoC Signal Represented by Pin	Direction (I/O)	Internal Pull-up (PU) or Pull-down (PD)	Power Rail
SMB_CLK1	As BIOS Starts	SMB_CLK1	I/O, OD	20K PU	V3P3S
	FUN_PIN_MUX Changed to 2	SPKR	O	20K PU	V3P3S
	SC_USE_SEL = 1	GPIOS_12	Set by SW	Design Specific	V3P3S
SMB_DATA2	As BIOS Starts	SMB_DATA2	I/O, OD	20K PU	V3P3S
	FUN_PIN_MUX Changed to 2	UART0_RXD	I	20K PU	V3P3S
	SC_USE_SEL = 1	GPIOS_13	Set by SW	Design Specific	V3P3S
SMB_CLK2	As BIOS Starts	SMB_CLK2	I/O, OD	20K PU	V3P3S
	FUN_PIN_MUX Changed to 2	UART0_TXD	O	20K PU	V3P3S
	SC_USE_SEL = 1	GPIOS_14	Set by SW	Design Specific	V3P3S
SATA_GP0	As BIOS Starts	SATA_GP0	I	20K PU	V3P3S
	SC_USE_SEL = 1	GPIOS_15	Set by SW	Design Specific	V3P3S
SATA_LEDN	As BIOS Starts	SATA_LEDN	O, OD	None	V3P3S
	SC_USE_SEL = 1	GPIOS_16	Set by SW	Design Specific	V3P3S
SATA3_GP0	As BIOS Starts	SATA3_GP0	I	20K PU	V3P3S
	SC_USE_SEL = 1	GPIOS_17	Set by SW	Design Specific	V3P3S
SATA3_LEDN	As BIOS Starts	SATA3_LEDN	O, OD	None	V3P3S
	SC_USE_SEL = 1	GPIOS_18	Set by SW	Design Specific	V3P3S
FLEX_CLK_SE0	Strap Sampling	0 = LPC / 1 = SPI	I	20K PU	V3P3S
	As BIOS Starts	FLEX_CLK_SE0	O	None	V3P3S
	SC_USE_SEL = 1	GPIOS_19	Set by SW	Design Specific	V3P3S
FLEX_CLK_SE1	Strap Sampling	0 = Reserved	I	20K PU	V3P3S
	As BIOS Starts	FLEX_CLK_SE1	O	None	V3P3S
	SC_USE_SEL = 1	GPIOS_20	Set by SW	Design Specific	V3P3S
ILB_SERIRQ	As BIOS Starts	ILB_SERIRQ	I/O	20K PU	V3P3S
	SC_USE_SEL = 1	GPIOS_29	Set by SW	Design Specific	V3P3S
PMU_RESETBUTTON_B	As BIOS Starts	PMU_RESETBUTTON_B	I	20K PU	V3P3S
	SC_USE_SEL = 1	GPIOS_30	Set by SW	Design Specific	V3P3S
AR51_RSVD	Strap Sampling	Strap: Reserved for Intel	I	None	V3P3S
	As BIOS Starts	AR51_RSVD	I/O	20K PU	V3P3S

Table 31-25. Signal Pins with Shared Functions - SUS Power Well (Sheet 1 of 3)

SoC Signal Pin Name	Functional Options and Occurrences	SoC Signal Represented by Pin	Direction (I/O)	Internal Pull-up (PU) or Pull-down (PD)	Power Rail
GPIO_SUS0	Strap Sampling	Strap: Reserved for Intel	I	20K PD	V3P3A
	As BIOS Starts	GPIO_SUS0	I	20K PU	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS0	Set by SW	Design Specific	V3P3A
GPIO_SUS1	Strap Sampling	Strap: 2.5 GbE Enable	I	20K PD	V3P3A
	NCSI Strap = 0	GPIO_SUS1	I	20K PU	V3P3A
	NCSI Strap = 1	NCSI_RXD0	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS1	Set by SW	Design Specific	V3P3A
GPIO_SUS2	Strap Sampling	Strap: Reserved for Intel	I	None	V3P3A
	As BIOS Starts	GPIO_SUS2	I	20K PU	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS2	Set by SW	Design Specific	V3P3A
CPU_RESET_B	Strap Sampling	Strap: Reserved for Intel	I	None	V3P3A
	As BIOS Starts	CPU_RESET_B	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS3	Set by SW	Design Specific	V3P3A
SUSPWRDNACK	As BIOS Starts	SUSPWRDNACK	O	None	VP3A
	SUS_USE_SEL = 1	GPIO_SUS4	Set by SW	Design Specific	V3P3A
PMU_SUSCLK	Strap Sampling	Strap: Reserved for Intel	I	20K PD	V3P3A
	As BIOS Starts	PMU_SUSCLK	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS5	Set by SW	Design Specific	V3P3A
PMU_PLTRST_B	Strap Sampling	Strap: Reserved for Intel	I	20K PD	V3P3A
	As BIOS Starts	PMU_PLTRST_B	O	None	V3P3A
PMU_SLP_DDRVTT_B	As BIOS Starts	PMU_SLP_DDRVTT_B	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS6	Set by SW	Design Specific	V3P3A
PMU_SLP_LAN_B	As BIOS Starts	PMU_SLP_LAN_B	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS7	Set by SW	Design Specific	V3P3A
PMU_WAKE_B	As BIOS Starts	PMU_WAKE_B	I	20K PU	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS8	Set by SW	Design Specific	V3P3A
PMU_PWRBTN_B	As BIOS starts	PMU_PWRBTN_B	I	20K PU	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS9	Set by SW	Design Specific	V3P3A
SUS_STAT_B	Strap Sampling	Strap: Reserved for Intel	I	20K PD	V3P3A
	As BIOS Starts	SUS_STAT_B	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS10	Set by SW	Design Specific	V3P3A
USB_OC0_B	As BIOS Starts	USB_OC0_B	I	20K PU	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS11	Set by SW	Design Specific	V3P3A
SPI_CS0_B [1]	Strap Sampling	Strap: AG3E Strap	I	20K PU	V3P3A
	As BIOS Starts	SPI_CS0_B	O	None	V3P3A
SPI_CS1_B	Strap Sampling	Strap: Reserved for Intel	I	20K PD	V3P3A
	As BIOS Starts	SPI_CS1_B	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS12	Set by SW	Design Specific	V3P3A

Table 31-25. Signal Pins with Shared Functions - SUS Power Well (Sheet 2 of 3)

SoC Signal Pin Name	Functional Options and Occurrences	SoC Signal Represented by Pin	Direction (I/O)	Internal Pull-up (PU) or Pull-down (PD)	Power Rail
GBE_EE_DI	As BIOS Starts	GBE_EE_DI	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS13	Set by SW	TBD	V3P3A
GBE_EE_DO	As BIOS Starts	GBE_EE_DO	I	20K PU	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS14	Set by SW	TBD	V3P3A
GBE_EE_SK	As BIOS Starts	GBE_EE_SK	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS15	Set by SW	TBD	V3P3A
GBE_EE_CS_N	As BIOS Starts	GBE_EE_CS_N	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS16	Set by SW	TBD	V3P3A
GBE_SDP0_0	As BIOS Starts	GBE_SDP0_0	I	20K PU	V3P3A
	After SW Sets	GBE_SDP0_0	Set by SW	TBD	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS17	Set by SW	TBD	V3P3A
GBE_SDP0_1	NCSI Strap = 0	GBE_SDP0_1	I	20K PU	V3P3A
	After SW Sets	GBE_SDP0_1	Set by SW	20K PU	V3P3A
	NCSI Strap = 1	NCSI_ARB_IN	I	SIP Controlled	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS18	Set by SW	TBD	V3P3A
GBE_LED0	As BIOS Starts	GBE_LED0	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS19	Set by SW	TBD	V3P3A
GBE_LED1	As BIOS Starts	GBE_LED1	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS20	Set by SW	TBD	V3P3A
GBE_LED2	As BIOS Starts	GBE_LED2	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS21	Set by SW	TBD	V3P3A
GBE_LED3	As BIOS Starts	GBE_LED3	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS22	Set by SW	TBD	V3P3A
GBE_SMBALRT_N	NCSI Strap = 0	GBE_SMBALRT_N	I/O, OD	20K PU	V3P3A
	NCSI Strap = 1	NCSI_CRS_DV	O	None	V3P3A
GBE_SMBCLK	NCSI Strap = 0	GBE_SMBCLK	I/O, OD	20K PU	V3P3A
	NCSI Strap = 1	NCSI_CLK_IN	I/O, OD	None	V3P3A
GBE_SMBD	NCSI Strap = 0	GBE_SMBD	I/O, OD	20K PU	V3P3A
	NCSI Strap = 1	NCSI_TX_EN	I	None	V3P3A
NCSI_RXD1	Strap Sampling	Strap: Ethernet during S5	I	20K PU	V3P3A
	NCSI Strap = 0	Reserved for Intel Use	n/a	None	V3P3A
	NCSI Strap = 1	NCSI_RXD1	O	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS23	Set by SW	TBD	V3P3A
GBE_MDIO0_I2C_CLK	As BIOS Starts	GBE_MDIO0_I2C_CLK	I/O, OD	20K PU	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS24	Set by SW	TBD	V3P3A
GBE_MDIO0_I2C_DATA	As BIOS Starts	GBE_MDIO0_I2C_DATA	I/O, OD	20K PU	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS25	Set by SW	TBD	V3P3A

Table 31-25. Signal Pins with Shared Functions - SUS Power Well (Sheet 3 of 3)

SoC Signal Pin Name	Functional Options and Occurrences	SoC Signal Represented by Pin	Direction (I/O)	Internal Pull-up (PU) or Pull-down (PD)	Power Rail
GBE_MDIO1_I2C_CLK	NCSI Strap = 0	GBE_MDIO1_I2C_CLK	I/O, OD	20K PU	V3P3A
	NCSI Strap = 1	NCSI_TXD1	I	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS26	Set by SW	TBD	V3P3A
GBE_MDIO1_I2C_DATA	NCSI Strap = 0	GBE_MDIO1_I2C_DATA	I/O, OD	20K PU	V3P3A
	NCSI Strap = 1	NCSI_TXD0	I	None	V3P3A
	SUS_USE_SEL = 1	GPIO_SUS27	Set by SW	TBD	V3P3A
NCSI_ARB_OUT	Strap Sampling	Strap: NCSI Strap	I	20K PD	V3P3A
	NCSI Strap = 0	Y59_RSVD	O	20K PU	V3P3A
	NCSI Strap = 1	NCSI_ARB_OUT	O	None	V3P3A

Notes:

1. If the GEN_PMCON1.RTC_PWR_STS bit, (PBASE + 0x20[2]) is:
   - 1'b1 then SoC will use the value of this hard pin strap to determine if it should stay in S5 or go to S0.
   - 1'b0 then SoC will use GEN_PMCON1.AFTERG3_EN bit to determine if it should stay in S5 or go to S0.

§ §

32 Signal Pin States and Termination

This chapter describes the states of each SoC signal during reset sequencing and the S5 (Soft-Off) power state. It also documents what signals have internal pull-up/pull-down/series termination resistors and their values.

32.1 Signal Pin States

Table 32-1. Reset State Definitions

Buffer Type	Buffer Description
High-Z	High-impedance state. For bi-directional signals (designated as I/O), external drivers are not expected.
Don't Care	The state of the input (driven or tri-stated) has no effect. For bi-directional signals (designated as I/O), it is assumed the output buffer is in a high-impedance state.
V_{OH}	Output voltage high.
V_{OL}	Output voltage low.
VOX-known	Output voltage known. The signal voltage level is defined by internal function configuration.
VOX-unknown	Output voltage unknown. The signal voltage level has an indeterminate value.
V_{IH}	Input voltage high.
V_{IL}	Input voltage low.
pull-up	This signal is pulled high by a pull-up resistor (internal or external).
pull-down	This signal is pulled low by a pull-down resistor (internal or external).
VIX-unknown	Input voltage unknown. The signal voltage level has an indeterminate value.
Running	The clock or signal is toggling because the function has not stopped.
Off	The power plane for this signal is powered down. Driver and Receiver buffers are turned off.

32.1.1 System Memory Signals

32.1.1.1 DDR3[0] Memory Signals

Table 32-2. System Memory Signals (DDR3[0])

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
DDR3_0_DQ[63:0]	I/O	High-Z	High-Z	High-Z
DDR3_0_MA[15:0]	O	High-Z	VOL	High-Z
DDR3_0_DQS[7:0] DDR3_0_DQSECC	I/O	High-Z	High-Z	High-Z
DDR3_0_DQSB[7:0] DDR3_0_DQSBECC	I/O	High-Z	High-Z	High-Z
DDR3_0_CK[3:0]	O	High-Z	VOH	High-Z
DDR3_0_CKB[3:0]	O	High-Z	VOL	High-Z
DDR3_0_CKE[3:0]	O	High-Z	VOL	VOL
DDR3_0_CSB[3:0]	O	High-Z	VOH	High-Z
DDR3_0_ODT[3:0]	O	High-Z	High-Z	High-Z
DDR3_0_RASB	O	High-Z	VOH	High-Z
DDR3_0_CASB	O	High-Z	VOH	High-Z
DDR3_0_WEB	O	High-Z	VOH	High-Z
DDR3_0_BS[2:0]	O	High-Z	VOL	High-Z
DDR3_0_DRAM_PWROK	I	VIH	VIH	High-Z
DDR3_0_DRAMRSTB	I/O			
DDR3_0_VCCA_PWROK	I	VIH	VIH	High-Z
DDR3_0_VREF	I	Analog	Analog	High-Z
DDR3_0_ODTPU	I/O			
DDR3_0_BS[2:0]	O	High-Z	VOL	High-Z
DDR3_0_DQPU	I/O			
DDR3_0_CMDPU	I/O			
DDR3_0_MON1P	I/O			
DDR3_0_MON1N	I/O			
DDR3_0_MON2P	I/O			
DDR3_0_MON2N	I/O			
DDR3_0_REFP DDR3_0_REFN	I	Running	Running	High-Z
DDR3_0_DQECC[7:0]	I/O	High-Z	High-Z	High-Z

32.1.1.2 DDR3[1] Memory Signals

Table 32-3. System Memory Signals (DDR3[1])

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
DDR3_1_DQ[63:0]	I/O	High-Z	High-Z	High-Z
DDR3_1_MA[15:0]	O	High-Z	VOL	High-Z
DDR3_1_DQS[7:0] DDR3_1_DQSECC	I/O	High-Z	High-Z	High-Z
DDR3_1_DQSB[7:0] DDR3_1_DQSBECC	I/O	High-Z	High-Z	High-Z
DDR3_1_CK[3:0]	O	High-Z	VOH	High-Z
DDR3_1_CKB[3:0]	O	High-Z	VOL	High-Z
DDR3_1_CKE[3:0]	O	High-Z	VOL	VOL
DDR3_1_CSB[3:0]	O	High-Z	VOH	High-Z
DDR3_1_ODT[3:0]	O	High-Z	High-Z	High-Z
DDR3_1_RASB	O	High-Z	VOH	High-Z
DDR3_1_CASB	O	High-Z	VOH	High-Z
DDR3_1_WEB	O	High-Z	VOH	High-Z
DDR3_1_BS[2:0]	O	High-Z	VOL	High-Z
DDR3_1_DRAM_PWROK	I	VIH	VIH	High-Z
DDR3_1_DRAMRSTB	I/O			
DDR3_1_VCCA_PWROK	I	VIH	VIH	High-Z
DDR3_1_VREF	I/O	Analog	Analog	High-Z
DDR3_1_ODTPU	I/O			
DDR3_1_BS[2:0]	O	High-Z	VOL	High-Z
DDR3_1_DQPU	I/O			
DDR3_1_CMDPU	I/O			
DDR3_1_MON1P	I/O			
DDR3_1_MON1N	I/O			
DDR3_1_MON2P	I/O			
DDR3_1_MON2N	I/O			
DDR3_1_REFP DDR3_1_REFN	I	Running	Running	High-Z
DDR3_1_DQECC[7:0]	I/O	High-Z	High-Z	High-Z

32.1.2 Thermal Management Signals

Table 32-4. Thermal Management Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
THERMTRIP_N	O, OD	High-Z	High-Z	Off
PROCHOT_B	I/O, OD	Pull-up	Pull-up	Off
MEMHOT_B	I	Pull-up	Pull-up	Off

32.1.3 SVID Interface Signals

Table 32-5. SVID Interface Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
SVID_ALERT_B	I	Pull-up	Pull-up	Off
SVID_DATA	I/O, OD	Pull-up	Pull-up	Off
SVID_CLK	O, OD	Pull-up	Pull-up	Off

32.1.4 Core Misc Signals

Table 32-6. Core Misc Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
NMI	I	Pull-down	Pull-down	Off
ERROR2_B	O	VOL	VOH	Off
ERROR1_B	O	VOL	VOH	Off
ERROR0_B	O	VOL	VOH	Off
IERR_B	O	VOL	VOH	Off
MCERR_B	O	VOL	VOH	Off
UART1_RXD	I	Pull-down	Pull-down	Off
UART1_TXD	O	VOL	VOH	Off
SMB_CLK0	I/O, OD	Pull-up	Pull-up	Off
SMB_DATA0	I/O, OD	Pull-up	Pull-up	Off
SMBALRT_N0	I/O, OD	Pull-up	Pull-up	Off
SMB_DATA1	I/O, OD	Pull-up	Pull-up	Off
SMB_CLK1	I/O, OD	Pull-up	Pull-up	Off
SMB_DATA2	I/O, OD	Pull-up	Pull-up	Off
SMB_CLK2	I/O, OD	Pull-up	Pull-up	Off

32.1.5 SATA/eSATA GEN2 Interface Signals

Table 32-7. SATA2 Interface Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
SATA_TXP[3:0]	O	Analog	Analog	Off
SATA_TXN[3:0]	O	Analog	Analog	Off
SATA_RXP[3:0]	I	Analog	Analog	Off
SATA_RXN[3:0]	I	Analog	Analog	Off
SATA_REFCLKP	I	High-Z	Running	Off
SATA_REFCLKN	I	High-Z	Running	Off
SATA_OBSP	O	Analog	Analog	Off
SATA_OBSN	O	Analog	Analog	Off
SATA_GP0	I	Pull-up	Pull-up	Off
SATA_LEDN	O, OD	High-Z	High-Z	Off

32.1.6 SATA3 Interface Signals

Table 32-8. SATA3 Interface Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
SATA3_TXP[1:0]	O	Analog	Analog	Off
SATA3_TXN[1:0]	O	Analog	Analog	Off
SATA3_RXP[1:0]	I	Analog	Analog	Off
SATA3_RXN[1:0]	I	Analog	Analog	Off
SATA3_REFCLKP	I	High-Z	Running	Off
SATA3_REFCLKN	I	High-Z	Running	Off
SATA3_OBSP	O	Analog	Analog	Off
SATA3_OBSN	O	Analog	Analog	Off
SATA3_GP0	I	Pull-up	Pull-up	Off
SATA3_LEDN	O, OD	High-Z	High-Z	Off

32.1.7 PCI Express Root Port Signals

Table 32-9. PCI Express Root Port Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
PCIE_TXP[15:0]	O	Analog	Analog	Off
PCIE_TXN[15:0]	O	Analog	Analog	Off
PCIE_RXP[15:0],	I	Analog	Analog	Off
PCIE_RXN[15:0]	I	Analog	Analog	Off
PCIE_REFCLKP	I	High-Z	Running	Off
PCIE_REFCLKN	I	High-Z	Running	Off
PCIE_OBSP	O	Analog	Analog	Off
PCIE_OBSN	O	Analog	Analog	Off
FLEX_CLK_SE0	O	VOL	Running	Off
FLEX_CLK_SE1	O	VOL	Running	Off

32.1.8 GbE Interface Signals

Refer to section 2 of the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*.

Table 32-10. GbE Interface Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5	External PU/PD	Input to the Core if NC-SI Unused
NCSI_TXD0	I	High-Z				0x0
NCSI_TXD1	I	High-Z				0x0
NCSI_RXD0	O	High-Z			Pull-up if NC-SI is set to a multi-drop configuration.	
NCSI_RXD1	O	High-Z			Pull-up if NC-SI is set to a multi-drop configuration.	
NCSI_CLK_IN	I/O, OD	High-Z				0x0
NCSI_TX_EX	I	High-Z				0x0
NCSI_CRS_DV	O	High-Z			Pull up if NC-SI is set to a multi-drop configuration.	
NCSI_ARB_IN	I	High-Z	If NC-SI hardware arbitration is disabled via the NC-SI ARB Enable EEPROM bit, the NCSI_ARB_IN pin is pulled-up internally.			
NCSI_ARB_OUT	O		If the device is in device off, the output should be stable high.			
GBE_TXP[3:0]	O	Analog	Analog	Analog		
GBE_TXN[3:0]	O	Analog	Analog	Analog		
GBE_RXP[3:0]	I	Analog	Analog	Analog		
GBE_RXN[3:0]	I	Analog	Analog	Analog		
GBE_REFCLKP	I	High-Z	Running	Running		
GBE_REFCLKN	I	High-Z	Running	Running		
GBE_OBSP	O	Analog	Analog	Analog		
GBE_OBSN	O	Analog	Analog	Analog		
GBE_SMBD	I/O, OD	Pull-up	Pull-up	Pull-up		
GBE_SMBCLK	I/O, OD	Pull-up	Pull-up	Pull-up		
GBE_SMBALRT_N	I/O, OD	Pull-up	Pull-up	Pull-up		
GBE_SDP0_0	I/O	Pull-up	Pull-up	Pull-up		
GBE_SDP0_1	I/O	Pull-up	Pull-up	Pull-up		
GBE_LED0	O	VOL	VOL	VOL		
GBE_LED1	O	VOL	VOL	VOL		
GBE_LED2	O	VOL	VOL	VOL		
GBE_LED3	O	VOL	VOL	VOL		
GBE_MDIO0_I2C_CLK	I/O, OD	Pull-up	Pull-up	Pull-up		
GBE_MDIO0_I2C_DATA	I/O, OD	Pull-up	Pull-up	Pull-up		
GBE_MDIO1_I2C_CLK	I/O, OD	Pull-up	Pull-up	Pull-up		
GBE_MDIO1_I2C_DATA	I/O, OD	Pull-up	Pull-up	Pull-up		

32.1.9 EEPROM Signals

Refer to Section 2 of the *Intel® Atom™ Processor C2000 Product Family Integrated GbE Controller Programmer's Reference Manual (PRM)*.

Table 32-11. EEPROM Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
GBE_EE_DI	O	VOL	VOL	VOL
GBE_EE_DO	I	Pull-up	Pull-up	Pull-up
GBE_EE_SK	O	VOL	VOL	VOL
GBE_EE_CS_N	O	VOL	VOL	VOL

32.1.10 Low Pin Count (LPC) Signals

Table 32-12. Low Pin Count (LPC) Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
LPC_AD[3:0]	I/O	Pull-up	Pull-up	Off
LPC_FRAMEB	O	VOH	VOH	Off
LPC_CLKOUT0	O	VOL	Running	Off
LPC_CLKOUT1	O	VOL	Running	Off
LPC_CLKRUNB	I/O OD	High-Z	High-Z	Off

32.1.11 Intel Legacy Block (ILB) Signals

Table 32-13. ILB Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
ILB_SERIRQ	I/O	Pull-up	Pull-up	Off

32.1.12 RTC Well Signals

Table 32-14. RTC Well Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
RTEST_B	I	VIX-unknown	VIH	VIL
RSMRST_B	I	VIX-unknown	VIH	VIL
COREPWROK	I	VIX-unknown	VIH	VIL
SRTCRST_B	I	VIX-unknown	VIH	VIL
BRTCX1_PAD	I/O	Running	Running	Running
BRTCX2_PAD	I/O	Running	Running	Running
BVCCRTC_EXTPAD	I/O	Analog	Analog	Analog

32.1.13 GPIO SUS Signals

Table 32-15. GPIO SUS Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
GPIO_SUS0	I/O	Pull-up	Pull-up	Pull-up
GPIO_SUS1	I/O	Pull-up	Pull-up	Pull-up
GPIO_SUS2	I/O	Pull-up	Pull-up	Pull-up
CPU_RESET_B	O	VOL	VOH	VOL

32.1.14 Power Management Unit (PMU) Interface

Table 32-16. PMU Interface Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
SUSPWRDNACK	O	VOL	VOH	VOX-unknown
PMU_SUSCLK	O	VOL	High-Z	High-Z
PMU_SLP_DDRVTT_B	O	VOL	VOH	VOL
PMU_SLP_S45_B	O	VOL	VOH	VOH
PMU_SLP_S3_B	O	VOL	VOH	VOL
PMU_SLP_LAN_B/GPIO_SUS7 (non-functional, always logic high)	O	VOH	VOH	VOH
PMU_WAKE_B	I	Pull-up	Pull-up	Pull-up
PMU_PWRBTN_B	I	Pull-up	Pull-up	Pull-up
PMU_RESETBUTTON_B	I	Pull-up	Pull-up	Off
PMU_PLTRST_B	O	VOL	VOH	VOL
SUS_STAT_B	O	VOH	VOH	VOL

32.1.15 USB2 Interface Signals

Table 32-17. USB2 Interface Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
USB_DP[3:0]	I/O	Analog	Analog	High-Z
USB_DN[3:0]	I/O	Analog	Analog	High-Z
USB_REFCLKP	I	High-Z	Running	Running
USB_REFCLKN	I	High-Z	Running	Running
USB_RCOMPO	O	Analog	Analog	Analog
USB_RCOMPI	I	Analog	Analog	Analog
USB_OBSP	O	Analog	Analog	Analog
USB_OC0_B	I	Pull-up	Pull-up	Pull-up

32.1.16 SPI and Flash Memory Signals

Table 32-18. SPI and Flash Memory Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
SPI_MISO	I	Pull-up	Pull-up	Pull-up
SPI_MOSI	I/O	Pull-down	VOL	Pull-up
SPI_CLK	O	VOL	VOL	Pull-up
SPI_CS0_B	O	VOH	VOH	Pull-up
SPI_CS1_B	O	VOH	VOH	Pull-up

32.1.17 GPIO DFX Signals

Table 32-19. GPIO DFX Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
DFX_PORT_CLK[1:0]	I/O	VOL	VOL	Off
DFX_PORT[15:0]	I/O	VOL	VOL	Off
CTBTRIGINOUT	I/O	Pull-up	Pull-up	Off
CTBTRIGOUT	O	VOL	VOL	Off
RCOMP_CORE_LVT	O	Analog	Analog	Off

32.1.18 CLK Interface

Table 32-20. CLK Receiver Interface

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
HPLL_REFP	I	Running	Running	Off
HPLL_REFN	I	Running	Running	Off
CLK14_IN	I	Pull-up	Pull-up	Off

32.1.19 JTAG and Debug Signals

Table 32-21. JTAG and Debug Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
TCK	I	Pull-down	Pull-down	Pull-down
TRST_B	I	Pull-up	Pull-up	Pull-up
TMS	I	Pull-up	Pull-up	Pull-up
TDI	I	Pull-up	Pull-up	Pull-up
TDO	O	VOL	VOL	VOL
CX_PRDY_B	O, OD	Pull-up	Pull-up	Off
CX_PREQ_B	I	Pull-up	Pull-up	Off

32.1.20 General-Purpose I/O Signals

Table 32-22. General-Purpose I/O Signals

Signal Name	Direction	During Reset (PWROK)	Post-Reset	S5
GPIO_SUS0	I/O	Pull-up	Pull-up	Pull-up
GPIO_SUS2	I/O	Pull-up	Pull-up	Pull-up

32.2 Integrated Termination Resistors

Table 32-23. Integrated Termination Resistors (Sheet 1 of 2)

Signal	Direction	Internal Termination Resistor Type	Nominal Value (Ω)
PROCHOT_B	I/O, OD	Pull-up	2K
MEMHOT_B	I	Pull-up	2K
SVID_DATA	I/O, OD	Pull-up	2K
SVID_CLK	O, OD	Pull-up	2K
SVID_ALERT_B	I	Pull-up	2K
NMI	I	Pull-down	20K
UART1_RXD	I	Pull-down	20K
SMB_CLK0	I/O, OD	Pull-up	20K
SMB_DATA0	I/O, OD	Pull-up	20K
SMBALRT_N0	I/O, OD	Pull-up	20K
SMB_DATA1	I/O, OD	Pull-up	20K
SMB_CLK1	I/O, OD	Pull-up	20K
SMB_DATA2	I/O, OD	Pull-up	20K
SMB_CLK2	I/O, OD	Pull-up	20K
SATA_GP0	I	Pull-up	20K
SATA3_GP0	I	Pull-up	20K
GBE_SMBD	I/O, OD	Pull-up	20K
GBE_SMBCLK	I/O, OD	Pull-up	20K
GBE_SMBALRT_N	I/O, OD	Pull-up	20K
GBE_EE_DO	I	Pull-up	20K
GBE_SDP0_0	I/O	Pull-up	20K
GBE_SDP0_1	I/O	Pull-up	20K
GBE_MDIO0_I2C_CLK	I/O, OD	Pull-up	20K
GBE_MDIO0_I2C_DATA	I/O, OD	Pull-up	20K
GBE_MDIO1_I2C_CLK	I/O, OD	Pull-up	20K
GBE_MDIO1_I2C_DATA	I/O, OD	Pull-up	20K
ILB_SERIRQ	I/O	Pull-up	20K
GPIO_SUS0	I/O	Pull-up	20K
GPIO_SUS1	I/O	Pull-up	20K
GPIO_SUS2	I/O	Pull-up	20K
PMU_WAKE_B	I	Pull-up	20K
PMU_PWRBTN_B	I	Pull-up	20K
PMU_RESETBUTTON_B	I	Pull-up	20K
USB_OC0_B	I	Pull-up	20K
SPI_MISO	I	Pull-up	20K
SPI_MOSI	I/O	Pull-down	20K
SPI_CLK	O	Pull-up	20K
SPI_CS0_B	O	Pull-up	20K
SPI_CS1_B	O	Pull-up	20K
CTBTRIGINOUT	I/O	Pull-up	20K
CLK14_IN	I	Pull-up	20K

Table 32-23. Integrated Termination Resistors (Sheet 2 of 2)

Signal	Direction	Internal Termination Resistor Type	Nominal Value (Ω)
TCK	I	Pull-down	2K
TRST_B	I	Pull-up	2K
TMS	I	Pull-up	2K
TDI	I	Pull-up	2K
CX_PRDY_B	O, OD	Pull-up	2K
CX_PREQ_B	I	Pull-up	2K

32.3 Strap Signals

Some of the signal pins are also used at power-up time as hardware-strapping pins. These pins are described in Section 16.2, "Pin-Based (Hard) Straps" on page 357.

32.4 Reserved Signals and Signals Not Used by Platform Board

All signals described as Reserved must be connected to the platform board as indicated in the signal description section. Some Reserved signals must be terminated on the platform board and others must have No Connection (NC) to them. See Table 31-23, "Reserved Signals" on page 611.

Unless specified otherwise in the signal description, connecting a Reserved or NC signal to a board supply voltage, VSS, or to any other signal pin, including each other, results in component malfunction or incompatibility with future processors.

For reliable operation, ensure the following:

- Connect unused input-signal pins and bi-directional-signal pins to the appropriate high- or low-signal level.
- Unused active-high inputs are connected through a resistor to ground (VSS).
- Unused outputs are left unconnected; however, this interferes with some Test Access Port (TAP) functions, complicate debug probing, and prevent boundary-scan testing.
- A resistor must be used when tying bi-directional signals to power or ground.
- When tying any unused signal to power or ground, using a resistor instead of a direct connection, allows for system testability.
- Find additional termination guidance in the *Intel® Atom™ Processor C2000 Product Family - Platform Design Guide (PDG).*

§ §

33 Signal Electrical and Timing Characteristics

Note: This chapter contains information that is subject to change.

This chapter is organized by signal interface. Each sub-chapter contains the interface DC, AC, and signal timing requirements and characteristics. Some of these requirements and characteristics are based on industry and Intel standards. When the SoC interface complies to a standard, the reader is referred to the standard for the requirements and characteristics. SoC exceptions to the standard, if any, are shown as well as any of the standard's optional interface characteristics as implemented by the SoC design.

Most of the information presented here is in Table and Figure form. In some cases, the reader is referred to one of the functional-description chapters where additional interface-related information is available.

33.1 DDR3 Memory Interface

33.1.1 DC Specifications

The DC and AC characteristics of the SoC DDR3 memory interface allow it to interface to and control SDRAM components complying to the *DDR3 SDRAM Specification JESD79-3E*. Both single-ended and differential signals are covered in the specification. Refer to the specification's sections:

- JESD79-3E, Section 8 - AC and DC Input Measurement Levels
- JESD79-3E, Section 9 - AC and DC Output Measurement Levels

The key DC and AC parameters for the SoC DDR3 memory controllers are shown in Table 33-1, Table 33-2 on page 632, Table 33-3 on page 634, and Table 33-4 on page 639.

Table 33-1. DDR3 and DDR3L Signal DC Specifications (Sheet 1 of 2)

Symbol	Parameter	Min	Typ	Max	Unit	Note
I_{IL}	Input Leakage Current	-	20	-	uA	10
Data Signals						
V_{IL}	Input Low Voltage	-	-	SMREF - 0.125	V	2, 3
V_{IH}	Input High Voltage	SMREF + 0.125	-	-	V	2, 4, 5
R_{ON}	DDR3L Data Buffer On Resistance	26	-	40	Ω	6
Reference Clock Signals, Command, and Data Signals						
V_{OL}	Output Low Voltage	-	(VDDQ/2)* (R_{ON} / (R_{ON}+R_{VTT_TERM}))	-	V	2, 7

Table 33-1. DDR3 and DDR3L Signal DC Specifications (Sheet 2 of 2)

Symbol	Parameter	Min	Typ	Max	Unit	Note
V_{OH}	Output High Voltage	-	VDDQ - ((VDDQ / 2)* (R_{ON}/ (R_{ON}+R_{VTT_TERM}))	-	V	2, 5, 7
Reference Clock Signals						
R_{ON}	DDR3L Clock Buffer On Resistance	26	-	40	Ω	6
Command Signals						
R_{ON}	DDR3 Command Buffer On Resistance	18	-	32	Ω	6
R_{ON}	DDR3 Reset Buffer On Resistance	-	-	40	Ω	6
V_{OL}	Output Low Voltage, signals: DDR3_0_DRAMRSTB DDR3_1_DRAMRSTB	-	-	0.2*VDDQ	V	1, 2
V_{OH}	Output High Voltage, signals: DDR3_0_DRAMRSTB DDR3_1_DRAMRSTB	0.9*VDDQ	-	-	V	1, 2
Control Signals						
R_{ON}	DDR3 Control Buffer On Resistance	18	-	32	Ω	6
DDR3 Miscellaneous Signals						
V_{IL}	Input Low Voltage, signals: DDR3_0_DRAM_PWROK, DDR3_1_DRAM_PWROK	-	-	0.55*VDDQ - 0.2	V	2, 3, 10
V_{IH}	Input High Voltage, signals: DDR3_0_DRAM_PWROK, DDR3_1_DRAM_PWROK	0.55*VDDQ + 0.2	-	-	V	2, 4, 5, 10

Notes:
1. Unless otherwise noted, all specifications in this table apply to all supported SDRAM frequencies.
2. Voltage rail VDDQ is 1.50V or 1.35V nominal depending on the voltage of all DIMMs connected to the SoC.
3. V_{IL} is the maximum voltage level at a receiving agent that will be interpreted as a logical low value.
4. V_{IH} is the minimum voltage level at a receiving agent that will be interpreted as a logical high value.
5. V_{IH} and V_{OH} may experience excursions above VDDQ. However, input signal drivers must comply with the signal quality specifications.
6. This is the pull-down driver resistance. Refer to processor signal integrity models for I/V characteristics. Reset drive does not have a termination.
7. R_{VTT_TERM} is the termination on the DIMM and not controlled by the SoC. Refer to the applicable UDIMM/SODIMM datasheet.
8. COMP resistance must be provided on the system board with 1% resistors. DDR_RCOMP resistors are terminated to VSS.
9. Input leakage current is specified for all DDR3 signals.
10. DDR3_0_DRAM_PWROK and DDR3_1_DRAM_PWROK must have a maximum of 15-ns rise or fall time over VDDQ * 0.55± 200 mV and the edge must be monotonic.

33.1.2 AC Specifications

33.1.2.1 DDR3 1333 MT/s

Table 33-2. DDR3 Signal AC Characteristics at 1333 MT/s (Sheet 1 of 2)

Symbol	Parameter	Min	Typ	Max	Unit	Note
Latency Timings						
tCL – tRCD – tRP	CAS Latency – RAS to CAS Delay – Pre-charge Command Period	9 - 9 - 9 10 - 10 - 10			tCK	
Electrical Characteristics						
T_{SLR_D}	DDR3_DQ[63:0], DDR3_DQS[7:0] DDR3_DQSB[7:0] Input Slew Rate	6.5		2.0	V/ns	
Clock Timings						
T_{CK}	DDR3_CK Period			1.5	ns	
T_{CH}	DDR3_CK High Time	0.7			ns	
T_{CL}	DDR3_CK Low Time	0.7			ns	
T_{SKEW}	Skew Between Any System Memory Differential Clock Pair (DDR3_CK/DDR3_CKB)			100	ps	
Command Signal Timings						
T_{CMD_CO}	DDR3_RASB, DDR3_CASB, DDR3_WEB, DDR3_MA[15:0], DDR3_BS[2:0] Edge placement accuracy	-145		145	ps	1
T_{CMD_VA}	Data, DDR3_DQ[63:0] and DDR3_DM[7:0] data available after command	500			ps	1
T_{CMD_VB}	Data, DDR3_DQ[63:0] and DDR3_DM[7:0] available before the command	520			ps	1
Control Signal Timings						
T_{CTRL_CO}	DDR3_CSB[1:0], DDR3_CKE[1:0], DDR3_ODT[1:0] Edge placement accuracy	-145		145	ps	2
T_{CTRL_VA}	Data, DDR3_DQ[63:0] and DDR3_DM[7:0] data available after control signal	500			ps	2
T_{CTRL_VB}	Data, DDR3_DQ[63:0] and DDR3_DM[7:0] available before the control signal	520			ps	2

Table 33-2. DDR3 Signal AC Characteristics at 1333 MT/s (Sheet 2 of 2)

Symbol	Parameter	Min	Typ	Max	Unit	Note
Data and Strobe Signal Timings						
$T_{DVA} + T_{DVB}$	Data, DDR3_DQ[63:0] and DDR3_DM[7:0] timing window available at the interface output for write commands. tDVB is data available before strobe and tDVA is data available after corresponding slope	475			ps	3
$T_{SU} + T_{HD}$	DDR3_DQ Input Setup plus Hold Time to DDR3_DQSB Rising or Falling Edge	220			ps	4
T_{DQS_CK}	DQS Edge Placement Accuracy to DDR3_CK Rising Edge Adjustable Range	-250		250	ps	
T_{WPRE}	DDR3_DQSB/DDR3_DQS Write Preamble Duration	1.0			TCK	
T_{WPST}	DDR3_DQSB/DDR3_DQS Write Postamble Duration	0.5			TCK	

Notes:
1. The CMD time is measured w.r.t. differential crossing of CK and CKB. The tCMDVB and tCMDVA are adjusted for proper CMD setup and hold time requirement at DRAM. The command timing assumes CMD-1N mode.
2. The CTL time is measured w.r.t. differential crossing of CK and CKB. The tCTLVB and tCTLVA are adjusted for proper CTL setup and hold time requirement at DRAM.
3. The accurate strobe placement using write training algorithm is performed which guarantees the required Data setup/ hold time w.r.t. strobe differential crossing at the DRAM input.
4. The read training algorithm places the DQS internally inside the DDR interface to have equal tSU and tHD timings.
5. All the timing windows are measured at 50% of the respective DDR signal swing.

33.1.2.2 DDR3 1600 MT/s

Table 33-3. DDR3 Signal AC Characteristics at 1600 MT/s (Sheet 1 of 2)

Symbol	Parameter	Min	Typ	Max	Unit	Note
Latency Timings						
tCL – tRCD – tRP	CAS Latency – RAS to CAS Delay – Pre-charge Command Period	11 - 11 - 11			tCK	
Electrical Characteristics						
T_{SLR_D}	DDR3_DQ[63:0], DDR3_DQS[7:0] DDR3_DQSB[7:0] Input Slew Rate	6.5		2.0	V/ns	
Clock Timings						
T_{CK}	DDR3_CK Period	1.25			ns	
T_{CH}	DDR3_CK High Time			0.5	ns	
T_{CL}	DDR3_CK Low Time			0.5	ns	
T_{SKEW}	Skew Between Any System Memory Differential Clock Pair (DDR3_CK/DDR3_CKB)	100.00			ps	
Command Signal Timings						
T_{CMD_CO}	DDR3_RASB, DDR3_CASB, DDR3_WEB, DDR3_MA[15:0], DDR3_BS[2:0] Edge placement accuracy	-145		145	ps	1
T_{CMD_VA}	Data, DDR3_DQ[63:0] and DDR3_DM[7:0] data available after command	425			ps	1
T_{CMD_VB}	Data, DDR3_DQ[63:0] and DDR3_DM[7:0] available before the command	400			ps	1
Control Signal Timings						
T_{CTRL_CO}	DDR3_CSB[1:0], DDR3_CKE[1:0], DDR3_ODT[1:0] Edge placement accuracy	-145		145	ps	2
T_{CTRL_VA}	Data, DDR3_DQ[63:0] and DDR3_DM[7:0] data available after control signal	425			ps	2
T_{CTRL_VB}	Data, DDR3_DQ[63:0] and DDR3_DM[7:0] available before the control signal	400			ps	2
Data and Strobe Signal Timings						
$T_{DVA} + T_{DVB}$	Data, DDR3_DQ[63:0] and DDR3_DM[7:0] timing window available at the interface output for write commands. tDVB is data available before strobe and tDVA is data available after corresponding slope	395			ps	3

Table 33-3. DDR3 Signal AC Characteristics at 1600 MT/s (Sheet 2 of 2)

Symbol	Parameter	Min	Typ	Max	Unit	Note
$T_{SU} + T_{HD}$	DDR3_DQ Input Setup plus Hold Time to DDR3_DQSB Rising or Falling Edge	170			ps	4
T_{DQS_CK}	DQS_DN Edge Placement Accuracy to DDR3_CK Rising Edge Adjustable Range	-125		125	ps	
T_{WPRE}	DDR3_DQSB/DDR3_DQS Write Preamble Duration			1.00	TCK	
T_{WPST}	DDR3_DQSB/DDR3_DQS Write Postamble Duration		0.30	0.50	TCK	

Notes:
1. The CMD time is measured w.r.t. differential crossing of CK and CKB. The tCMDVB and tCMDVA are adjusted for proper CMD setup and hold time requirement at DRAM. The command timing assumes CMD-1N mode.
2. The CTL time is measured w.r.t. differential crossing of CK and CKB. The tCTLVB and tCTLVA are adjusted for proper CTL setup and hold time requirement at DRAM.
3. The accurate strobe placement using write training algorithm is performed which guarantees the required data setup/ hold time w.r.t. strobe differential crossing at the DRAM input.
4. The read training algorithm places the DQS internally inside the DDR interface to have equal tSU and tHD timings.
5. All the timing windows are measured at 50% of the respective DDR signal swing.

33.1.3 Interface Timing Parameters and Waveforms

Figure 33-1. Electrical Test Circuit Diagram

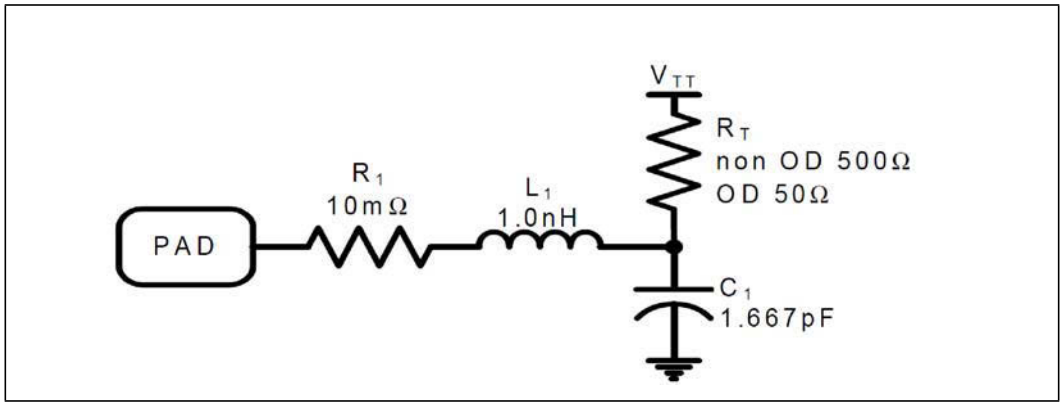


Figure 33-2. DDR3 Command / Control and Clock Timing Diagram

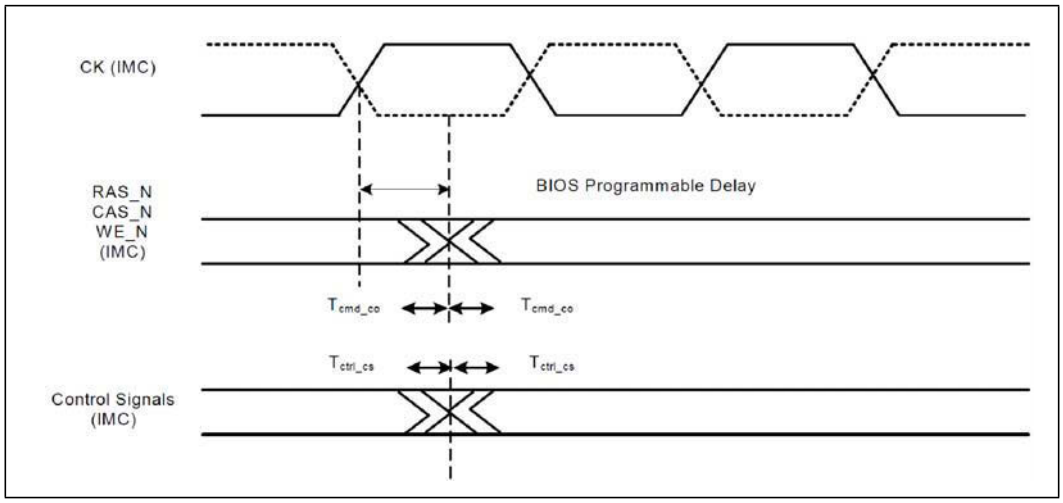


Figure 33-3. DDR3 Clock to Output Timing Diagram

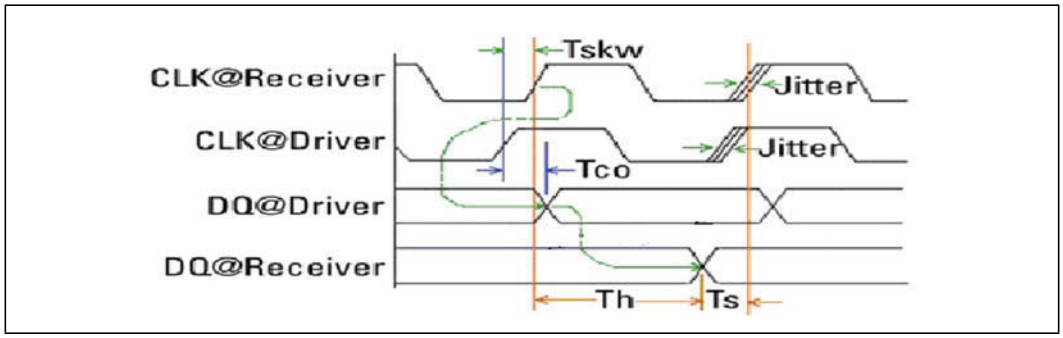
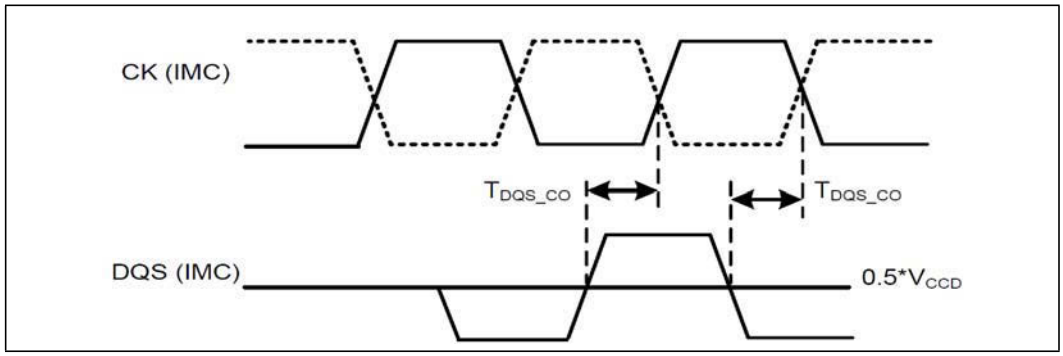


Figure 33-4. DDR3 Clock to DQS_DN Skew Timing Diagram



33.1.4 DDR3 Signal Quality Specifications

Various scenarios for the DDR3 Signals have been simulated to generate a set of layout guidelines which are available in the *Intel® Atom™ Processor C2000 Product Family Platform Design Guide* (PDG).

Overshoot (or undershoot) is the absolute value of the maximum voltage above or below V_{SS}. The overshoot/undershoot specifications limit transitions beyond specified maximum voltages or V_{SS} due to the fast signal edge rates. The processor can be damaged by single and/or repeated overshoot or undershoot events on any input, output, or I/O buffer if the charge is large enough (i.e., if the over/undershoot is great enough). Baseboard designs which meet signal integrity and timing requirements and which do not exceed the maximum overshoot or undershoot limits listed in Table 33-4 on page 639 ensures reliable I/O performance for the lifetime of the processor.

33.1.4.1 Overshoot/Undershoot Magnitude

Magnitude describes the maximum potential difference between a signal and its voltage reference level. For the processor, both are referenced to V_{SS}. It is important to note that the overshoot and undershoot conditions are separate and their impact must be determined independently.

The pulse magnitude and duration must be used to determine if the overshoot/undershoot pulse is within specifications.

33.1.4.2 Overshoot/Undershoot Pulse Duration

Pulse duration describes the total amount of time that an overshoot/undershoot event exceeds the overshoot/undershoot reference voltage. The total time could encompass several oscillations above the reference voltage. Multiple overshoot/undershoot pulses within a single overshoot/undershoot event may need to be measured to determine the total pulse duration.

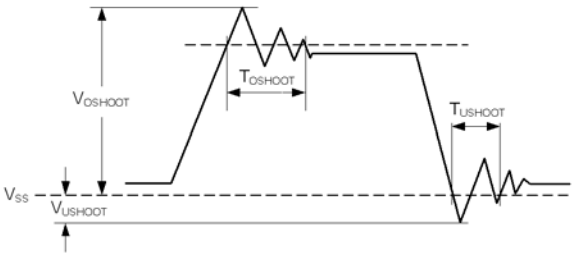
Note: Oscillations below the reference voltage cannot be subtracted from the total overshoot/undershoot pulse duration.

Table 33-4. DDR3 I/O Overshoot and Undershoot Specifications

Parameter	Symbol	Max	Units	Fig	Notes
Overshoot Voltage	V_{OSHOOT}	1.2 * VDDQ	V	33-5	1, 2
Overshoot Duration	T_{OSHOOT}	0.2 * T_{CH}Min	ns	33-5	1, 3
Undershoot Voltage	V_{USHOOT}	0.1 * VDDQ	V	33-5	1, 2
Undershoot Duration	T_{USHOOT}	0.2 * T_{CH}Min	ns	33-5	1, 3

Notes:
1. Values are measured at the SoC component pin/ball.
2. VDDQ is a SoC voltage supply group. See Table 34-3, "Voltage Supply Requirements Under Normal Operating Conditions" on page 690.
3. T_{CH}Min is the minimum value specified for CLK High Time T_{CH}. See Table 33-2, "DDR3 Signal AC Characteristics at 1333 MT/s" on page 632 and Table 33-3, "DDR3 Signal AC Characteristics at 1600 MT/s" on page 634.

Figure 33-5. Maximum Acceptable Overshoot/Undershoot Diagram



33.1.5 Other DDR3 Controller Electrical Specifications

Besides the signals covered in JESD79-3E, the SoC DDR3 Memory Controllers have additional interface signals.

The SoC electrical requirements for the DDR3 differential reference clock inputs are in Section 33.16.1, "Host, DDR3, PCI Express, SATA2 Reference Clocks" on page 668.

The DC parameters for the Power-OK input signals are shown in Table 33-5. These signals are:

- DDR3_0_DRAM_PWROK
- DDR3_1_DRAM_PWROK
- DDR3_0_VCCA_PWROK
- DDR3_1_VCCA_PWROK

Table 33-5. DDR3 Power OK Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	-	0.3	V	For all DDR3 speeds
V_{IH}	Input High Voltage	1.1	-	V	

33.2 PCI Express Root Port Interface

The SoC has up to 16 PCI Express* ports depending on product SKU. Each port consists of a Transmitter differential pair and a Receiver differential pair which are in the 1.0-Volt Core power well of the SoC.

- PCIE_TXP[15:0], PCIE_TXN[15:0] (Low Voltage Differential)
- PCIE_RXP[15:0], PCIE_RXN[15:0] (Low Voltage Differential)

See Section 4.3. Electrical Sub-Block of the *PCI Express Base Specification*, Revision 2.1 for DC and AC timing specifications for the host Transmitter and Receiver channels. The SoC supports devices with 5.0 GT/s and 2.5 GT/s capabilities.

The SoC electrical requirements for the PCIe* differential reference clock inputs are in Section 33.16.1, “Host, DDR3, PCI Express, SATA2 Reference Clocks” on page 668.

The SoC provides an integrated SMBus controller and interface that can be used in a PCI Express* interface design. The electrical characteristics are in Section 33.10, “SMBus 2.0 Interfaces” on page 659.

The SoC provides a Platform Reset output signal pin, PMU_PLTRST_B, and a PMU_WAKE_B input signal pin that can be used by the platform board design to create the PCIe interface signals PERST# and WAKE# respectively. The electrical characteristics for these signals are in Section 33.19, “SoC Reset and Power Management Unit (PMU) Interface” on page 677.

The SoC provides two general-purpose clock output pins from the SoC clock control unit. The signals are called FLEX_CLK_SE0 and FLEX_CLK_SE1. The electrical characteristics are in Section 33.17, “General Clocks Provided by SoC Interfaces” on page 674.

33.3 2.5 and 1 Gigabit Ethernet (GbE) Interface

For GbE reference clock input specifications, see Section 33.16.2, "GbE Reference Clock" on page 670.

The SoC has up to four GbE ports depending on product SKU. Each port consists of a Transmitter differential pair and a Receiver differential pair which are in the 1.0-Volt SUS power well of the SoC.

- GBE_TXP[3:0], GBE_TXN[3:0] (Low Voltage Differential)
- GBE_RXP[3:0], GBE_RXN[3:0] (Low Voltage Differential)

The electrical specifications conform to standards and depend on the mode of operation:

- SGMII (MAC to PHY)
- 1000BASE-KX (1 GbE)
- 2500BASE-X (2.5 GbE)

33.3.1 SGMII (MAC to PHY)

The SoC is designed to support 1000BASE-T when an external, SGMII-capable Physical-Layer device (PHY) is used in the platform board design. The SoC interface with the PHY conforms with the industry's Serial Gigabit Media Independent Interface (SGMII) Specification. The specification is a modified Physical Coding Sublayer (PCS) layer of the Gigabit Media Independent Interface (GMII) of the *IEEE Standard 802.3*-2008*, SECTION THREE, Clause 36, type 1000BASE-X. Refer to Clause 36 as well as Clause 37 of the IEEE 802.3 standard. The interface circuitry conforms to the IEEE Standard 802.3*-2008, SECTION FIVE, Clause 70 - Physical Medium Dependent Sublayer and Baseband Medium, Type 1000BASE-KX. Refer to 70.7 1000BASE-KX electrical characteristics of Clause 70.

33.3.2 1000BASE-KX (1 GbE)

The SoC is designed to operate with Ethernet interface circuitry that conforms to of the *IEEE Standard 802.3*-2008*, Section 5, Clause 70 - Physical Medium Dependent Sublayer and Baseband Medium, Type 1000BASE-KX. Refer to 70.7 1000BASE-KX electrical characteristics of Clause 70.

33.3.3 2500BASE-X (2.5 GbE)

The SoC can operate at a 2.5 Gb/s data rate using 1000BASE-KX. This mode uses a 3.125-GHz clock rate and requires the GbE reference clock input to be 125 MHz instead of 100 MHz. See Section 33.16.2, "GbE Reference Clock" on page 670. For other electrical parameters refer to Section 33.3.2, "1000BASE-KX (1 GbE)" on page 642.

33.3.3.1 Transmitter Characteristics

Transmitter characteristics at Test Point 1 (TP1) are summarized in Table 33-6. The location of TP1 is shown in Figure 33-6 on page 644. The signaling speed shall be 3.125 GBd ±100 ppm. The corresponding unit interval is nominally 320 ps.

Table 33-6. Transmitter Characteristics

Parameter	Refer to	Value	Units
Signaling speed, per lane		3.125 ± 100 ppm	GBd
Differential peak-to-peak output voltage	Section 33.3.3.1.2	800 to 1200	mV
Differential peak-to-peak output voltage (max.) with TX disabled		30	mV
Common-mode voltage limits	Section 33.3.3.1.2	–0.4 to 1.9	V
Differential output return loss (min.)	Section 33.3.3.1.3	See the two equations in Section 33.3.3.1.3.	dB
Differential output template	Section 33.3.3.1.4	See Figure 33-9 on page 647 and Table 33-7 on page 648.	V
Transition time[1] (20%-80%)	Section 33.3.3.1.5	60 to 130	ps
Output jitter (max. peak-to-peak), Random jitter	Section 33.3.3.1.6	0.27	UI
Output jitter (max. peak-to-peak), Deterministic jitter	Section 33.3.3.1.6	0.17	UI
Output jitter (max. peak-to-peak), Total jitter[2]	Section 33.3.3.1.6	0.35	UI

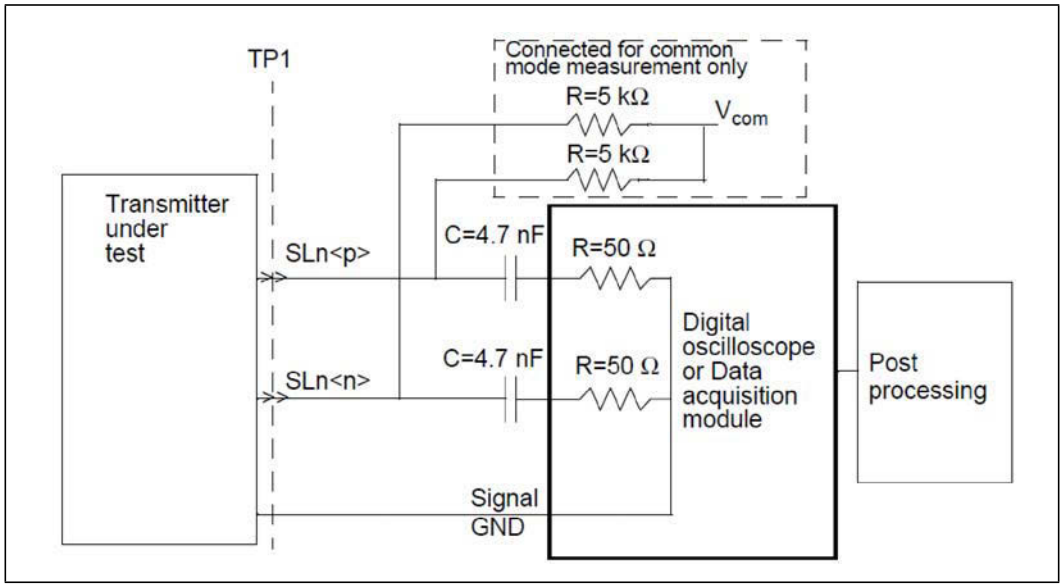
1. Transition time parameters are recommended values, not compliance values.
2. At BER 10^{-12}.

33.3.3.1.1 Test Fixtures

The test fixture of Figure 33-6, or its functional equivalent, is required for measuring the transmitter specifications described in Table 33-6 on page 643, with the exception of return loss.

The differential load impedance applied to the transmitter output by the test fixture depicted in Figure 33-6 shall be 100 Ω with a return loss greater than 20 dB from 100 MHz to 2000 MHz.

Figure 33-6. Transmit Test Fixture



33.3.3.1.2 Output Amplitude

While transmitting the test pattern specified in Annex 48A - Jitter Test Patterns of *IEEE Standard 802.3*-2008*,

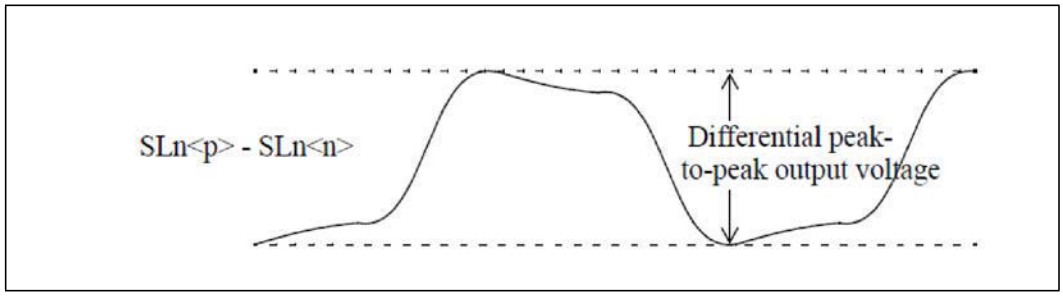
1. The transmitter maximum differential peak-to-peak output voltage shall be less than 1200 mV.
2. The minimum differential peak-to-peak output voltage shall be greater than 800 mV.
3. The maximum difference between any two lanes' differential peak-to-peak output voltage shall be less than or equal to 150 mV.

See Figure 33-7 for an illustration of the definition of differential peak-to-peak output voltage.

DC-referenced voltage levels are not defined since the receiver is AC-coupled. The common-mode voltage of SLn<p> and SLn<n> shall be between –0.4V and 1.9V with respect to signal ground as measured at V_{com} in Figure 33-6.

Note: SLn<p> and SLn<n> are the positive and negative sides of the differential signal pair for Lane n (n = 0,1,2,3).

Figure 33-7. Transmitter Differential Peak-to-Peak Output Voltage Definition



33.3.3.1.3 Output Return Loss

For frequencies from 100 MHz to 2000 MHz, the differential return loss, in dB with *f* in MHz, of the transmitter shall meet the requirements of the two equations:

$$\text{ReturnLoss}(f) \geq 10 \quad \text{for } 100\ \text{MHz} \leq f < 625\ \text{MHz}$$

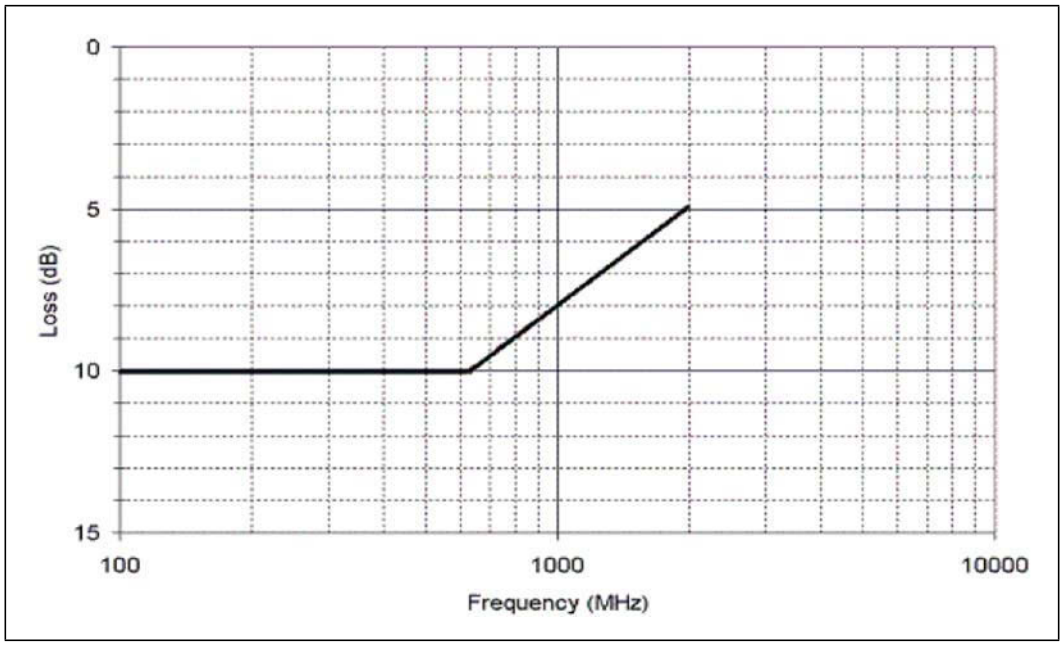
and

$$\text{ReturnLoss}(f)\ 10 - 10 \times \log\left(f / 625\right) \quad \text{for } 625\ \text{MHz} \leq f \leq 2000\ \text{MHz}$$

This output impedance requirement applies to all valid output levels. The reference impedance for differential return loss measurements shall be 100 Ω.

The minimum differential output return loss is shown in Figure 33-8.

Figure 33-8. Minimum Differential Output Return Loss



33.3.3.1.4 Differential Output Template

The transmitter differential output signal is defined at TP1 as shown in Figure 33-6 on page 644 and Figure 33-7 on page 645. The transmitter shall provide equalization such that the output waveform falls within the template shown in Figure 33-9 on page 647 for the test pattern specified in 48A.2 of Annex 48A - Jitter Test Patterns of *IEEE Standard 802.3*-2008*, with all other transmitters active. All other transmitters shall be terminated with a load meeting the requirements described in Section 33.3.3.1.1, "Test Fixtures" on page 644. Voltage and time coordinates for inflection points on Figure 33-9 on page 647 are given in Table 33-7 on page 648. The waveform under test shall be normalized by using the following procedure:

a. Align the output waveform under test, to achieve the best fit along the horizontal time axis.
b. Calculate the +1 low frequency level as V_{lowp} = average of any two successive unit intervals (2UI) between 2.5 UI and 5.5 UI.
c. Calculate the 0 low frequency level as V_{lowm} = average of any two successive unit intervals (2UI) between 7.5 UI and 10.5 UI.
d. Calculate the vertical offset to be subtracted from the waveform as $V_{off} = (V_{lowp} + V_{lowm}) / 2$.
e. Calculate the vertical normalization factor for the waveform as $V_{norm} = (V_{lowp} - V_{lowm}) / 2$.
f. Calculate the normalized waveform as:
Normalized_Waveform = (Original_Waveform – V_{off}) × (0.69/V_{norm}).
g. Align the Normalized_Waveform under test, to achieve the best fit along the horizontal time axis.

Figure 33-9. Normalized Transmit Template

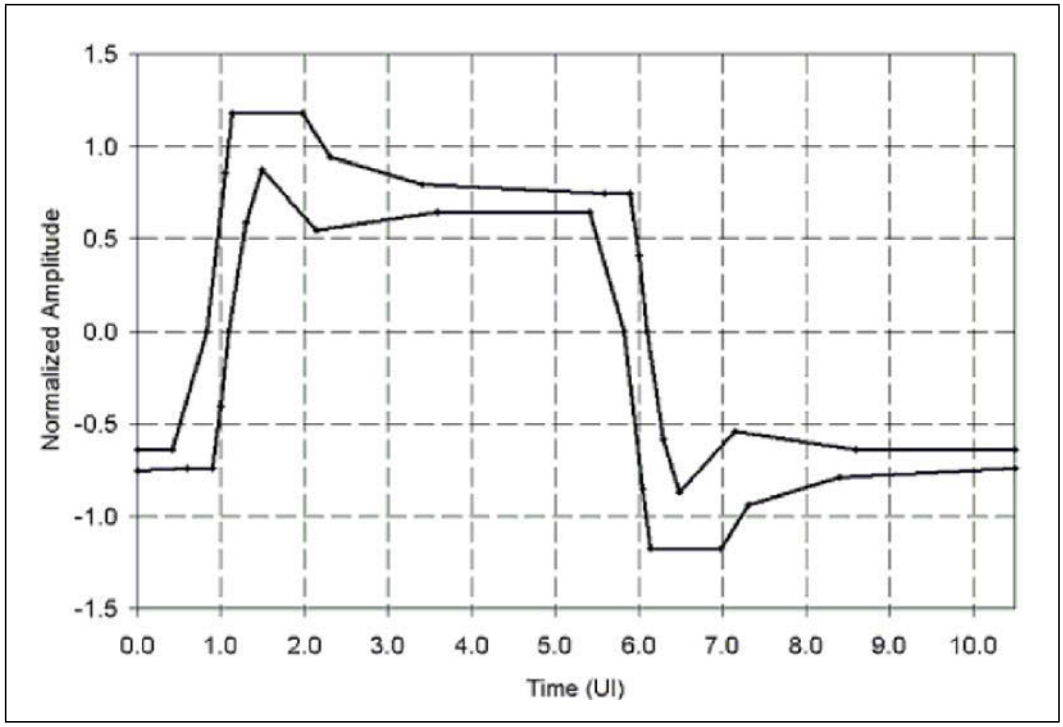


Table 33-7. Normalized Transmit Time Domain Template

Upper Limit				Lower Limit			
Time (UI)	Amplitude	Time (UI)	Amplitude	Time (UI)	Amplitude	Time (UI)	Amplitude
0.000	–0.640	5.897	0.740	0.000	–0.754	5.409	0.640
0.409	–0.640	5.997	0.406	0.591	–0.740	5.828	0.000
0.828	0.000	6.094	0.000	0.897	–0.740	6.050	–0.856
1.050	0.856	6.294	–0.586	0.997	–0.406	6.134	–1.175
1.134	1.175	6.491	–0.870	1.094	0.000	6.975	–1.175
1.975	1.175	7.141	–0.546	1.294	0.586	7.309	–0.940
2.309	0.940	8.591	–0.640	1.491	0.870	8.500	–0.790
3.409	0.790	10.500	–0.640	2.141	0.546	10.500	–0.742
5.591	0.740			3.591	0.640		

33.3.3.1.5 Transition Time

The rising edge transition time is recommended to be between 60 ps and 130 ps as measured at the 20% and 80% levels of the peak-to-peak differential value of the waveform using the high-frequency test pattern of 48A.1 in Annex 48A - Jitter Test Patterns of *IEEE Standard 802.3*-2008*. The falling edge transition time is recommended to be between 60 ps and 130 ps as measured at the 80% and 20% levels of the peak-to-peak differential value of the waveform using the high-frequency test pattern of 48A.1.

33.3.3.1.6 Transmit Jitter

The transmitter shall have a maximum total jitter of 0.350 UI peak-to-peak, a maximum deterministic component of 0.170 UI peak-to-peak, and a maximum random component of 0.270 UI peak-to-peak. Jitter specifications include all but 10–12 of the jitter population. Transmit jitter test requirements are specified in Section 33.3.3.1.7.

33.3.3.1.7 Transmit Jitter Test Requirements

Transmit jitter is defined with respect to the transmitter differential output signal at TP1, as shown in Figure 33-6 on page 644 and Figure 33-9 on page 647, and the test procedure resulting in a BER bathtub curve such as that described in Annex 48B - Jitter Test Methods of *IEEE Standard 802.3*-2008*. For the purpose of jitter measurement, the effect of a single-pole high-pass filter with a 3 dB point at 1.875 MHz is applied to the jitter. The data pattern for jitter measurements shall be the jitter tolerance test pattern defined in Annex 48A.5. For this test, all other transmitters shall be active and terminated with a load meeting the requirements described in Section 33.3.3.1.1, "Test Fixtures" on page 644. Crossing times are defined with respect to the mid-point (0 V) of the AC-coupled differential signal.

33.3.3.2 Receiver Characteristics

Receiver characteristics at TP4 are summarized in Table 33-8 and detailed in Section 33.3.3.2.1 through Section 33.3.3.2.1

Table 33-8. Receiver Characteristics

Parameter	Refer to	Value	Units
Bit error ratio	Section 33.3.3.2.1	10^{-12}	
Signaling speed, per lane	Section 33.3.3.2.2	3.125 ± 100 ppm	GBd
Unit interval (UI) nominal	Section 33.3.3.2.2	320	ps
Receiver coupling	Section 33.3.3.2.3	AC	
Differential input peak-to-peak amplitude (maximum)	Section 33.3.3.2.4	1600	mV
Differential input return loss[1] (minimum)	Section 33.3.3.2.5	See the two equations in Section 33.3.3.1.3, "Output Return Loss" on page 646.	dB

1. Relative to 100 Ω differential.

33.3.3.2.1 Receiver Interference Tolerance

The receiver interference tolerance shall be measured as described in Annex 69A - Interference Tolerance Testing of the *IEEE Standard 802.3*-2008*, with the parameters specified in Table 33-9. The data pattern for the interference tolerance test shall be the continuous jitter test pattern as defined in 48A.5 in Annex 48A - Jitter Test Patterns of *IEEE Standard 802.3*-2008*. The receiver shall satisfy the requirements for interference tolerance specified in Annex 69A.

Table 33-9. Interference Tolerance Parameters

Parameter	Value	Units
Target BER	10^{-12}	
m_{TC}[1] (min.)	1.0	
Amplitude of broadband noise (min. RMS)	8.1	mV
Applied transition time (20%–80%, min.)	130	ps
Applied sinusoidal jitter (min. peak-to-peak)	0.17	UI
Applied random jitter (min. peak-to-peak)[2]	0.18	UI
Applied duty cycle distortion (min. peak-to-peak)	0.0	UI

1. m_{TC} is defined in Equation 69A–6 of Annex 69A in the *IEEE Standard 802.3*-2008*.
2. Applied random jitter is specified at a BER of 10^{-12}.

33.3.3.2.2 Signaling Speed

The signaling speed shall be 3.125 GBd ±100 ppm. The corresponding unit interval is nominally 320 ps.

33.3.3.2.3 AC-Coupling

The receiver shall be AC-coupled to the backplane to allow for maximum interoperability between various 10 Gb/s components. AC-coupling is considered to be part of the receiver for the purposes of this specification unless explicitly stated otherwise. It should be noted that there may be various methods for AC-coupling in actual implementations.

Note: It is recommended that the maximum value of the coupling capacitors be limited to 4.7 nF. This will limit the inrush currents to the receiver that could damage the receiver circuits when repeatedly connected to transmit modules with a higher voltage level.

33.3.3.2.4 Input Signal Amplitude

Receivers shall accept differential input signal peak-to-peak amplitudes produced by compliant transmitters connected without attenuation to the receiver, and still meet the BER requirement specified in Section 33.3.3.2.1, "Receiver Interference Tolerance" on page 650. Note that this may be larger than the 1200 mV differential maximum of Section 33.3.3.1.2, "Output Amplitude" on page 645 due to the actual transmitter output and receiver input impedances. The input impedance of a receiver can cause the minimum signal into a receiver to differ from that measured when the receiver is replaced with a 100 Ω test load. Since the channel is AC-coupled, the absolute voltage levels with respect to the receiver ground are dependent on the receiver implementation.

33.3.3.2.5 Differential Input Return Loss

For frequencies from 100 MHz to 2000 MHz, the differential return loss, in dB with *f* in MHz, of the receiver shall be greater than or equal to the two equations in Section 33.3.3.1.3, "Output Return Loss" on page 646. This return loss requirement applies to all valid input levels. The reference impedance for differential return loss measurements is 100 Ω.

33.4 Network Controller MDIO Interface

There are two external-PHY Management Channels. Each is a two-wire I^2C channel operating in Standard Mode. The SoC is designed to interface with circuitry conforming to the *IEEE Standard 802.3*-2008* standards for MDIO.

The SoC signals are in the 3.3V SUS power well of the SoC.

External PHY Management Channel 0

- GBE_MDIO0_I2C_CLK (Input / Output-OD)
- GBE_MDIO0_I2C_DATA (Input / Output-OD)

External PHY Management Channel 1

- GBE_MDIO1_I2C_CLK (Input / Output-OD)
- GBE_MDIO1_I2C_DATA (Input / Output-OD)

For electrical specifications, refer to the *IEEE Standard 802.3*-2008* standard, Clause 22 and Annex 45A (informative) Clause 45 MDIO electrical interface.

33.5 Network Controller Sideband Interface (NC-SI)

The SoC is designed to interface with circuitry conforming to the specifications in Section 10.2 Electrical and Signal Characteristics and Requirements, of the *DSP0222 Network Controller Sideband Interface (NC-SI) Specification*, Version: 1.0.0.

http://www.dmtf.org/sites/default/files/standards/documents/DSP0222_1.0.0.pdf

These signals are in the 3.3V SUS power well of the SoC.

- NCSI_TX_EN (Input)
- NCSI_CLK_IN (Input/Output OD)
- NCSI_CRS_DV (Output)
- NCSI_ARB_IN (Input)
- NCSI_ARB_OUT (Input)
- NCSI_TXD0 (Input)
- NCSI_TXD1 (Input)
- NCSI_RXD0 (Output)
- NCSI_RXD1 (Output)

For electrical specifications, refer to Section 10 of the *DSP0222 Network Controller Sideband Interface (NC-SI) Specification*, Version: 1.0.0.

33.6 Network Controller EEPROM Interface

33.6.1 DC Specifications

Table 33-10 contains the DC specifications for the GbE EEPROM interface signals. These signals are in the 3.3V SUS power well of the SoC.

- GBE_EE_DI (Output)
- GBE_EE_DO (Input)
- GBE_EE_SK (Output)
- GBE_EE_CS (Output)

Table 33-10. GbE EEPROM Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	V3P3A+0.3	V	
V_{OL}	Output Low Voltage	–	0.4	V	@ 4 mA
V_{OH}	Output High Voltage	2.4	V3P3A	V	@ –4 mA

33.6.2 Interface Timing Parameters and Waveforms

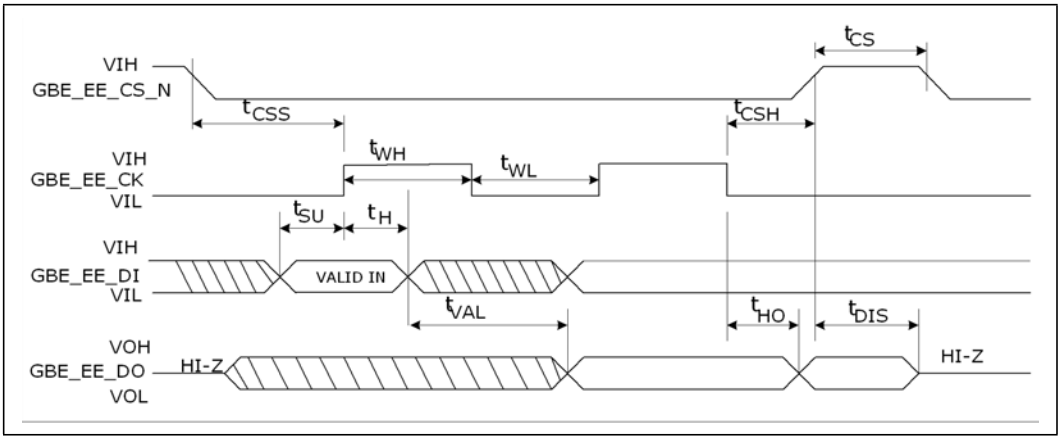
Table 33-11 contains the timing specifications for the GbE EEPROM interface signals.

Table 33-11. GbE EEPROM Signal Timing Specifications

Symbol	Parameter	Min	Typ	Max	Units	Figure
t_{SCK}	GBE_EE_SK Clock Frequency [1, 2]	0	2	2.1	MHz	
t_{RI}	Input Rise Time			2	µs	
t_{FI}	Input Fall Time			2	µs	
t_{WH}	GBE_EE_SK High Time	200	250		ns	33-10
t_{WL}	GBE_EE_SK Low Time	200	250		ns	33-10
t_{CS}	GBE_EE_CS_N (active-low signal) High Time	250			ns	33-10
t_{CSS}	GBE_EE_CS_N (active-low signal) Setup Time	250			ns	33-10
t_{CSH}	GBE_EE_CS_N (active-low signal) Hold Time	250			ns	33-10
t_{SU}	Data-In Setup Time	50			ns	33-10
t_{H}	Data-In Hold Time	50			ns	33-10
t_{VAL}	Output Valid	0		200	ns	33-10
t_{HO}	Output Hold Time	0			ns	33-10
t_{DIS}	Output Disable Time			250	ns	33-10

Notes:
1. Clock is 2 MHz.
2. 50% duty cycle.

Figure 33-10. GbE EEPROM Timing Diagram



33.7 Network Controller Miscellaneous Interfaces

33.7.1 GbE SMBus 2.0 Interface

Table 33-12 contains the DC specifications for the GbE SMBus 2.0 interface signals. These signals are in the 3.3V SUS power well of the SoC.

- GBE_SMBCLK (Input / Output-OD)
- GBE_SMBD (Input / Output-OD)
- GBE_SMBALRT_N (Input / Output-OD)

Table 33-12. GbE SMBus 2.0 Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	V3P3A+0.3	V	
V_{OL}	Output Low Voltage	–	0.1 * V3P3A	V	Open Drain
I_{OL}	Output Low Current	–	4	mA	Open Drain
V_O	Output Voltage applied to pin	–	3.47	V	Open Drain

The GbE SMBus 2.0 interface timing is the same as the other SMBus 2.0 interface ports of the SoC. See Section 33.10.2, "Interface Timing Parameters and Waveforms" on page 660 for the timing specifications for the GbE SMBus 2.0 interface signals.

33.7.2 GbE LED and Software-Defined Pins (SDP)

Table 33-13 contains the DC specifications for the GbE SDP interface signals. These signals are in the 3.3V SUS power well of the SoC.

- GBE_LED0 (Output)
- GBE_LED1 (Output)
- GBE_LED2 (Output)
- GBE_LED3 (Output)
- GBE_SDP0_0 (Input / Output)
- GBE_SDP0_1 (Input / Output)

Table 33-13. GbE SDP Pin Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	V3P3A+0.3	V	
V_{OL}	Output Low Voltage	–	0.4	V	@ 4 mA
V_{OH}	Output High Voltage	2.4	V3P3A	V	@ –4 mA

33.8 SATA2 and SATA3 Controller Interfaces

The SoC has up to four SATA2 ports and two SATA3 ports depending on product SKU. Each port consists of a Transmitter differential pair and a Receiver differential pair which are in the 1.0-Volt Core power well of the SoC.

- SATA_TXP[3:0], SATA_TXN[3:0] (Low Voltage Differential)
- SATA_RXP[3:0], SATA_RXN[3:0] (Low Voltage Differential)
- SATA3_TXP[1:0], SATA3_TXN[1:0] (Low Voltage Differential)
- SATA3_RXP[1:0], SATA3_RXN[1:0] (Low Voltage Differential)

For Serial ATA (SATA) interface electrical specifications, refer to Tables 29–34 in Section 7.2 of the *Serial ATA Revision 3.0 Specification*. The SoC as a SATA Host supports Gen1i, Gen1m, Gen2i, Gen 2m, and Gen3i as defined in the specification. The SoC supports Gen1m and Gen2m External SATA (eSATA) on the SATA2 Controller only.

Nominal channel speeds are 1.5 (Gen1) and 3.0 (Gen2) Gbps for the SATA2 and SATA3 controllers. The SATA3 controller also supports 6.0 Gbps.

The SoC electrical requirements for the SATA2 and SATA3 differential reference clock inputs are in Section 33.16.1, “Host, DDR3, PCI Express, SATA2 Reference Clocks” on page 668.

Other SATA Controller signals not part of the *Serial ATA Revision 3.0 Specification*:

- SATA_GP0, SATA3_GP0 (Input)
- SATA_LEDN, SATA3_LEDN (OD Output)

Table 33-14. SATA GP0 Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	V3P3S+0.3	V	

Table 33-15. SATA LED Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{OL}	Output Low Voltage	-	0.1	V	Open Drain @ 4 mA
V_O	Output Voltage applied to pin	–	3.47	V	Open Drain

33.9 USB 2.0 Interface

The SoC has four USB 2.0 ports. Each port consists of a Transceiver differential pair which is in the 1.0-Volt SUS power well of the SoC.

- USB_DP[3:0], USB_DN[3:0] (Transceiver, Low Voltage Differential)

For USB 2.0 interface electrical specifications, refer to Section 7 of the *Universal Serial Bus Specification,* Revision 2.0. The SoC as a USB Host supports:

- Low-Speed Signaling Mode (1.5 Mb/s)
- Full-Speed Mode (12 Mb/s)
- High-Speed Mode (480 Mb/s)

The SoC electrical requirements for the USB differential reference clock inputs are in Section 33.16.1, "Host, DDR3, PCI Express, SATA2 Reference Clocks" on page 668.

Other USB Host signals not part of the *Universal Serial Bus Specification,* Revision 2.0:

- USB_OC0_B (Over Current Indicator input)

Table 33-16. USB Over-Current Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.3	0.8	V	Not 5-Volt Tolerant
V_{IH}	Input High Voltage	2.0	V3P3A+0.3	V	

33.10 SMBus 2.0 Interfaces

The electrical specifications for the SMBus that is part of the GbE controller are shown in Section 33.7, "Network Controller Miscellaneous Interfaces" on page 656.

33.10.1 DC Specifications

These SMBus 2.0 interface signals are in the 3.3V Core power well of the SoC:

SMBus 2.0 Unit 0 - Legacy, typically for DIMM SPD

- SMB_CLK0 (Input / Output-OD)
- SMB_DATA0 (Input / Output-OD)
- SMBALRT_N0 (Input / Output-OD)

SMBus 2.0 Unit 1 - Host, SMT

- SMB_CLK1 (Input / Output-OD)
- SMB_DATA1 (Input / Output-OD)

SMBus 2.0 Unit 2 - PECI

- SMB_CLK2 (Input / Output-OD)
- SMB_DATA2 (Input / Output-OD)

For DC electrical specifications, refer to Section 3.1.3, High-Power DC Specifications, of the *System Management Bus (SMBus) Specification,* Version 2.0. In Table 3 of the specification, the Maximum Nominal Bus Voltage (VDD) of the SoC SMBus 2.0 controllers is 3.47 Volts rather than the 5.5-Volt value shown in the specification.

When the controller is configured for I^2C mode, the SoC controller complies with the DC electrical specifications in Chapter 6 of the *I^2C-bus Specification and User Manual*, Rev. 03.

33.10.2 Interface Timing Parameters and Waveforms

This subsection contains the timing parameters for all four SMBus 2.0 units of the SoC:

- Unit 0 - Legacy, typically for DIMM SPD
  - As an SMBus 2.0 Master, provides SMB_CLK0 at 83 kHz.
  - Cannot be an SMB 2.0 Target.
  - Can be configured to be an I^2C bus Master, Standard Mode, provides SMB_CLK0 at 83 kHz.
- Unit 1 - Host, SMT
  - As an SMBus 2.0 Master, provides SMB_CLK1 at 80 kHz.
  - As an SMB 2.0 Target, can operate at 10 kHz -100kHz.
  - Can be configured to be an I^2C bus Master, Standard Mode, provides SMB_CLK1 at 100 kHz.
  - Can be configured to be an I^2C bus Master, Fast Mode, provides SMB_CLK1 at 400 kHz.
  - Can be configured to be an I^2C bus Master, Fast Mode Plus, provides SMB_CLK1 at 1000 kHz.
- Unit 2 - PECI
  - As an SMB 2.0 Target, can operate at 10 kHz -100 kHz.
  - Cannot be an SMBus 2.0 Master.
  - Cannot be configured to be an I^2C bus Master or Target.
- SMBus GbE - Ethernet Controller
  - As an SMBus 2.0 Master, provides GBE_SMBCLK at 84 kHz.
  - As an SMB 2.0 Target, can operate at 10 kHz -100 kHz.
  - Cannot be configured to be an I^2C bus Master or Target.

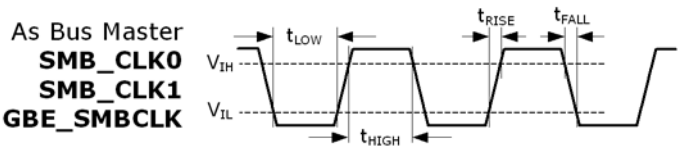
Table 33-17 and Figure 33-11 show the SMBus 2.0 clock output parameters when the SoC controller is the SMBus Master, and when it is an I^2C Master. The SoC output clock conforms with the T_{LOW}, T_{HIGH}, T_R, and T_F shown in the *System Management Bus (SMBus) Specification,* Version 2.0 and the *I^2C-bus Specification and User Manual*, Rev. 03 specifications. As a master, the SoC controllers comply with all other timing parameters defined by the specifications.

As an SMBus 2.0 or I^2C Target, each SoC controller conforms to the timing parameters defined by the specifications.

Table 33-17. When Bus Master - SMBus and I²C Output Clock Signal Timing Specifications

Symbol	Parameter	Min	Nominal	Max	Unit
f_{SMB}	SMB_CLK0 Frequency (SMBus 2.0)	–	83	–	kHz
	SMB_CLK0 Frequency (I^2C Standard Mode)	–	83	–	
	SMB_CLK1 Frequency (SMBus 2.0)	–	80	–	
	SMB_CLK1 Frequency (I^2C Standard Mode)	–	100	–	
	SMB_CLK1 Frequency (I^2C Fast Mode)	–	400	–	
	SMB_CLK1 Frequency (I^2C Fast Mode Plus)	–	1000	–	
	GBE_SMBCLK Frequency (SMBus 2.0)	–	84	–	

Figure 33-11. When Bus Master - SMBus and I²C Output Clock Signal Timing Drawing



33.11 Low Pin Count (LPC) Interface

The SoC contains an LPC port which includes two output clock signals and the serialized interrupt signal, ILB_SERIRQ. These signals are in the 3.3V Core power well of the SoC.

- LPC_CLKOUT[1:0] (Output)
- LPC_AD[3:0] (Input / Output)
- LPC_FRAMEB (Output)
- LPC_CLKRUNB (Input / Output-OD)
- ILB_SERIRQ (Input / Output)

For the SoC LPC interface electrical specifications, refer to Section 10, Electrical Specification, of the *Intel Low Pin Count (LPC) Interface Specification, Revision 1.1*. Sections 4, and 7 through 9 of the specification contain the timing parameters.

33.12 Serial Peripheral Interface (SPI) Bus Interface

33.12.1 DC Specifications

Table 33-18 contains the DC specifications for the SPI interface signals. These signals are in the 3.3V SUS power well of the SoC.

- SPI_MISO (Input)
- SPI_MOSI (Input / Output)
- SPI_CLK (Output)
- SPI_CS0_B (Output)
- SPI_CS1_B (Output)

Table 33-18. SPI Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	V3P3A+0.3	V	
V_{OL}	Output Low Voltage	–	0.4	V	@ 4 mA
V_{OH}	Output High Voltage	2.4	V3P3A	V	@ –4 mA

33.12.2 Interface Timing Parameters and Waveforms

Table 33-19, Table 33-20, and Figure 33-12 contain the timing specifications for the SPI interface signals.

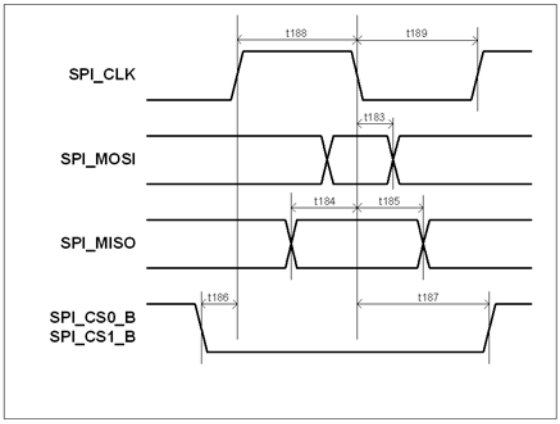
Table 33-19. SPI (33 MHz) Signal Timing Specifications

Symbol	Parameter	Min	Typ	Max	Units	Fig	Notes
t180	Serial Clock Frequency - 33 MHz Operation	–	33	–	MHz		
t183	Tco of SPI_MOSI with respect to serial clock falling edge at the host.	–5	–	5	ns	33-12	
t184	Setup of SPI_MISO with respect to serial clock falling edge at the host.	6	–	–	ns	33-12	
t185	Hold of SPI_MISO with respect to serial clock falling edge at the host.	0	–	–	ns	33-12	
t186	Setup of SPI_CSB[1:0] assertion with respect to serial clock rising at the host.	30	–	–	ns	33-12	
t187	Hold of SPI_CSB[1:0] deassertion with respect to serial clock falling at the host.	30	–	–	ns	33-12	
t188	SPI_SCLK High Time	14	–	–	ns	33-12	
t189	SPI_SCLK Low Time	14	–	–	ns	33-12	

Table 33-20. SPI (20 MHz) Signal Timing Specifications

Symbol	Parameter	Min	Typ	Max	Units	Fig	Notes
t180	Serial Clock Frequency - 20 MHz Operation	–	20	–	MHz	33-12	
t183	Tco of SPI_MOSI with respect to serial clock falling edge at the host.	–5		13	ns	33-12	
t184	Setup of SPI_MISO with respect to serial clock falling edge at the host.	6	–	–	ns	33-12	
t185	Hold of SPI_MISO with respect to serial clock falling edge at the host.	0	–	–	ns	33-12	
t186	Setup of SPI_CSB[1:0] assertion with respect to serial clock rising at the host.	30	–	–	ns	33-12	
t187	Hold of SPI_CSB[1:0] deassertion with respect to serial clock falling at the host.	30	–	–	ns	33-12	
t188	SPI_SCLK High Time	22	–	–	ns	33-12	
t189	SPI_SCLK Low Time	22	–	–	ns	33-12	

Figure 33-12. SPI Timing Diagram



Note: SPI_MISO - t184 and t185 are referencing clock edge where the SoC will sample the SPI_MISO pin. The slave transmitted this bit on the previous falling clock edge which is not shown.

33.13 High-Speed UART Interface

33.13.1 DC Specifications

Table 33-21 contains the DC specifications for the high-speed UART interface signals. These signals are in the 3.3V Core power well of the SoC.

- UART0_RXD (Input)
- UART1_RXD (Input)
- UART0_TXD (Output)
- UART1_TXD (Output)

Table 33-21. High-Speed UART Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	V3P3S+0.3	V	
V_{OL}	Output Low Voltage	–	0.4	V	@ 4 mA
V_{OH}	Output High Voltage	2.4	V3P3S	V	@ –4 mA

33.13.2 Interface Timing Parameters and Waveforms

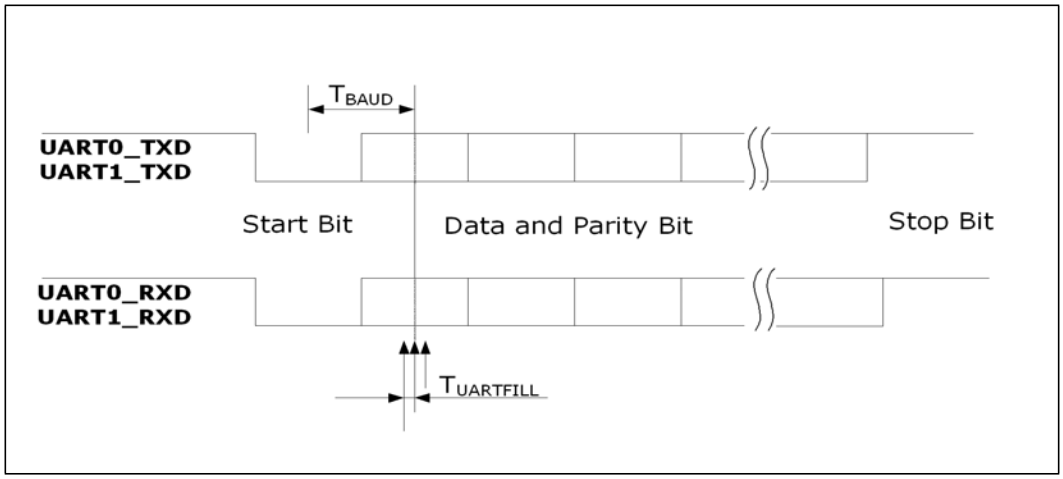
Table 33-22, and Figure 33-13 contain the timing specifications for the high-speed UART interface signals.

Table 33-22. High-Speed UART Signal Timing Specifications

Symbol	Parameter	Min	Max	Units	Fig	Notes
T_{RISE}	Rise Time	2.5	5	ns	33-13	1, 2
T_{FALL}	Fall Time	2.5	5	ns	33-13	1, 2
$T_{UARTFILL}$	UART Sampling Filter Period	20	—	—	33-13	3

Notes:
1. Based on the total trace length of 1-4” total maximum, capacitance of 27 pF and board impedance of 30–75 Ω.
2. Measured from 10–90%.
3. Each bit including start and stop bit is sampled three times at center of a bit at an interval of 20 ns (minimum). If three sampled values do not agree, then UART noise error is generated.

Figure 33-13.High-Speed UART Timing Diagram



33.14 Speaker Interface

33.14.1 DC Specifications

Table 33-23 contains the DC specifications for the Speaker signal. This signal is in the 3.3V Core power well of the SoC.

- SPKR (Output)

Table 33-23. Speaker Interface Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{OL}	Output Low Voltage	–	0.4	V	@ 4 mA
V_{OH}	Output High Voltage	2.4	V3P3S	V	@ –4 mA

33.15 Customer General-Purpose I/O (GPIO) Interfaces

33.15.1 DC Specifications

Table 33-24 and Table 33-25 contain the DC specifications for the Customer GPIO signals.

The following signals are in the 3.3V Core power well of the SoC. See Table 33-24.

- GPIOS_[30:0] (Input, Output)

Table 33-24. Customer GPIO - Core Power Well Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	V3P3S+0.3	V	
V_{OL}	Output Low Voltage	–	0.4	V	@ 4 mA
V_{OH}	Output High Voltage	2.4	V3P3S	V	@ –4 mA

The following signals are in the 3.3V SUS power well of the SoC. See Table 33-25.

- GPIO_SUS[27:0] (Input, Output)

Table 33-25. Customer GPIO - SUS Power Well Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	V3P3A+0.3	V	
V_{OL}	Output Low Voltage	–	0.4	V	@ 4 mA
V_{OH}	Output High Voltage	2.4	V3P3A	V	@ –4 mA

33.16 SoC Reference Clock Interfaces

33.16.1 Host, DDR3, PCI Express, SATA2 Reference Clocks

- HPLL_REF[P, N] - Host Reference Clock, Differential, Spread Spectrum
- DDR3_0_REF[P, N] - DDR3 Memory Controller Channel 0 Reference Clock, Differential, Spread Spectrum
- DDR3_1_REF[P, N] - DDR3 Memory Controller Channel 1 Reference Clock, Differential, Spread Spectrum
- PCIE_REFCLK[P, N] - PCI Express* Controller Reference Clock, Differential, Spread Spectrum
- SATA_REFCLK[P, N] - SATA2 Controller Reference Clock, Differential, Spread Spectrum

Table 33-26 shows the required clock period based on:

- PPM Tolerance = 35 ppm
- Cycle-to-Cycle Jitter = 85 ps
- Spread = -0.50%

Table 33-27 has the AC requirements for these reference clock inputs.

Table 33-26. Clock Period Requirements - Differential Input - Spread Spectrum

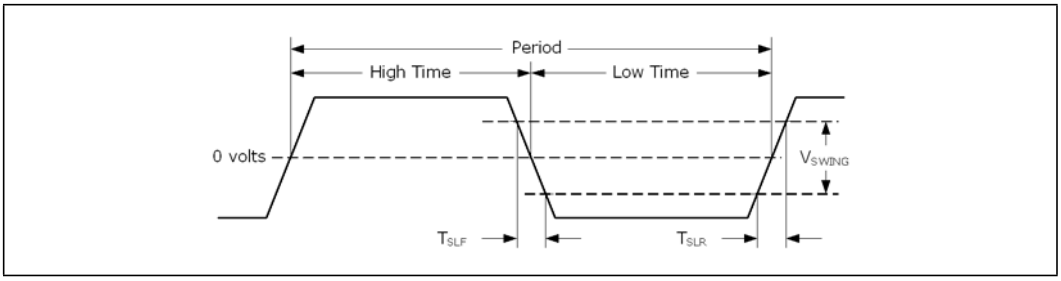
Center Frequency (MHz)	Measurement Window		Clock Period (ns)	Fig
100.00	1 Clock	- Clock-to-Clock Jitter AbsPer Min	9.88999	33-14
	1 µs	-SSC Short-Term Average Min	9.97499	
	0.1 sec	- ppm Long-Term Average Min	9.99999	
	0.1 sec	0 ppm Period Nominal	10.00000	
	0.1 sec	+ ppm Long-Term Average Max	10.00035	
	1 µs	+SSC Short-Term Average Max	10.02535	
	1 Clock	+ Clock-to-Clock Jitter AbsPer Max	10.11035	

Table 33-27. AC Electrical Requirements - Differential Input - Spread Spectrum

Parameter	Symbol	Type of Measurement	Min	Typ	Max	Units	Fig	Notes
Rising Edge Slew Rate	t_{SLR}	Differential Measurement	1.0	-	4	V/ns	33-14	1, 3
Falling Edge Slew Rate	t_{SLF}	Differential Measurement	1.0	-	4	V/ns	33-14	1, 3
Slew Rate Variation	t_{SLVAR}	Single-ended Measurement	-	-	20	%	-	2, 3
Maximum Output Voltage *includes overshoot*	V_{HIGH}	Single-ended Measurement	-	-	1150	mV	33-20	2
Minimum Output Voltage *includes undershoot*	V_{LOW}	Single-ended Measurement	-300	-	-	mV	33-20	2
Differential Voltage Swing	V_{SWING}	Differential Measurement	300	-	-	mV	33-14	1
Crossing Point Voltage	V_{XABS}	Single-ended Measurement	300	-	550	mV		2, 4
Crossing Point Variation	$V_{XABSVAR}$	Single-ended Measurement	-	-	140	mV		2, 5
Duty Cycle *Ratio of pulse High Time and the Period*	DCYC	Differential Measurement	45	-	55	%	33-14	1

Notes:
1. Differential Measurement - Measured from differential waveform on a component test board.
2. Single-ended Measurement - Measured from a single-ended waveform on a component test board.
3. Slew Rate is measured within the voltage range VSWING when centered at differential voltage = 0. In other words, it is measured from -150 mV to +150 mV on the differential waveform.
4. Vcross is defined at the voltage where Clock = Clock#.
5. Only applies to the differential rising edge (Clock rising, Clock# falling.)

Figure 33-14. Clock Period and Slew Rate Diagram - Differential Measurement



33.16.2 GbE Reference Clock

- GBE_REFCLK[P, N] - GbE Controller Reference Clock, Differential, no Spread Spectrum.

Table 33-28 shows the required clock period based on:

- Cycle-to-Cycle Jitter = 85 ps
- Spread = 0.00% (no spread spectrum)

Section 33-27, "AC Electrical Requirements - Differential Input - Spread Spectrum" on page 669 has the AC requirements for these reference clock inputs.

Table 33-28. Clock Period Requirements - Differential Input - No Spread Spectrum

Center Frequency (MHz)	Measurement Window		Clock Period (ns)	Fig
100.00	1 Clock	- Clock-to-Clock Jitter AbsPer Min	9.91499	33-14
	0.1 sec	- ppm Long-Term Average Min	9.99999	
	0.1 sec	0 ppm Period Nominal	10.00000	
	0.1 sec	+ ppm Long-Term Average Max	10.00001	
	1 Clock	+ Clock-to-Clock Jitter AbsPer Max	10.08501	
125.00	1 Clock	- Clock-to-Clock Jitter AbsPer Min	7.9149	
	0.1 sec	- ppm Long-Term Average Min	7.99999	
	0.1 sec	0 ppm Period Nominal	8.00000	
	0.1 sec	+ ppm Long-Term Average Max	8.00001	
	1 Clock	+ Clock-to-Clock Jitter AbsPer Max	8.08501	

33.16.3 SATA3 Reference Clock

- SATA3_REFCLK[P, N] - SATA3 Controller Reference Clock, Differential
  - — SoC can function with either Spread Spectrum or non Spread Spectrum reference clocks.

33.16.3.1 With Spread Spectrum

Table 33-26, "Clock Period Requirements - Differential Input - Spread Spectrum" on page 668 shows the required clock period based on:

- PPM Tolerance = 35 ppm
- Cycle-to-Cycle Jitter = 85 ps
- Spread = -0.50%

Table 33-27, "AC Electrical Requirements - Differential Input - Spread Spectrum" on page 669 has the AC requirements for these reference clock inputs.

33.16.3.2 With no Spread Spectrum

Table 33-29 shows the required clock period based on:

- PPM Tolerance = 35 ppm
- Cycle-to-Cycle Jitter = 85 ps
- Spread = 0.00% (no spread spectrum)

Table 33-27, "AC Electrical Requirements - Differential Input - Spread Spectrum" on page 669 has the AC requirements for these reference clock inputs.

Table 33-29. Clock Period Requirements - Differential Input - No Spread Spectrum

Center Frequency (MHz)	Measurement Window		Clock Period (ns)	Fig
100.00	1 Clock	- Clock-to-Clock Jitter AbsPer Min	9.91465	33-14
	0.1 sec	- ppm Long-Term Average Min	9.99965	
	0.1 sec	0 ppm Period Nominal	10.00000	
	0.1 sec	+ ppm Long-Term Average Max	10.00035	
	1 Clock	+ Clock-to-Clock Jitter AbsPer Max	10.08535	

33.16.4 USB 2.0 Reference Clock

- USB_REFCLK[P, N] - USB 2.0 Controller Reference Clock, Differential, no Spread Spectrum.

Table 33-30 shows the required clock period based on:

- PPM Tolerance = 35 ppm
- Cycle-to-Cycle Jitter = 250 ps
- Spread = 0.00% (no spread spectrum)

Table 33-27, "AC Electrical Requirements - Differential Input - Spread Spectrum" on page 669 has the AC requirements for these reference clock inputs.

Table 33-30. Clock Period Requirements - Differential Input - No Spread Spectrum

Center Frequency (MHz)	Measurement Window		Clock Period (ns)	Fig
96.00	1 Clock	- Clock-to-Clock Jitter AbsPer Min	10.16667	33-14
	0.1 sec	- ppm Long-Term Average Min	10.41667	
	0.1 sec	0 ppm Period Nominal	10.41667	
	0.1 sec	+ ppm Long-Term Average Max	10.41667	
	1 Clock	+ Clock-to-Clock Jitter AbsPer Max	10.66667	

33.16.5 14.318 MHz Reference Clock

- CLK14_IN - 14.318 MHz Reference Clock, single-ended SoC input, 3.3V Core-well input buffer

Table 33-31 contains the DC specifications and Table 33-32 contains the timing specifications.

Table 33-31. CLK14_IN Signal DC Specifications

Parameter	Symbol	Min	Typ	Max	Units	Fig
Input High Voltage	V_{IH}	2.0	-	V3P3S +0.3	V	33-19
Input Low Voltage	V_{IL}	-0.3	-	0.8	V	33-20
Maximum Output Voltage *includes overshoot*	V_{HIGH}	-	-	1150	mV	33-20
Minimum Output Voltage *includes undershoot*	V_{LOW}	-300	-	-	mV	33-20

Table 33-32. CLK14_IN Signal Timing Specifications

Parameter	Symbol	Conditions	Min	Typ	Max	Units	Fig
Clock Period	t_{PERIOD}	Measured at V_T	-	69.842	-	ns	33-21
Clock-to-Clock Jitter		Measured at V_T	-	-	1000	ps	33-21
Duty Cycle *Ratio of pulse High Time and the Period*	DCYC		45	-	55	%	33-21
Rising Edge Slew Rate	t_{SLR}	Measured from V_{TL} to V_{TH}	1	-	4	V/ns	33-21
Falling Edge Slew Rate	t_{SLF}	Measured from V_{TH} to V_{TL}	1	-	4	V/ns	33-21
Threshold Voltage - High	V_{TH}		2.0			V	33-21
Threshold Voltage - Middle	V_T		1.5			V	33-21
Threshold Voltage - Low	V_{TL}		0.8			V	33-21

33.17 General Clocks Provided by SoC Interfaces

33.17.1 DC Specifications

This signal is in the 3.3V SUS power well of the SoC. See Table 33-33. It is a single-ended clock signal.

- PMU_SUSCLK (Output)

Table 33-33. SUS Clock (RTC Clock) Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{OL}	Output Low Voltage	–	0.4	V	@ 4 mA
V_{OH}	Output High Voltage	2.4	V3P3A	V	@ –4 mA

These signals are in the 3.3V Core power well of the SoC. See Table 33-34. Each is a single-ended clock signal.

- FLEX_CLK_SE0 (Output)
- FLEX_CLK_SE1 (Output)

Table 33-34. Flex Clock Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{OL}	Output Low Voltage	–	0.4	V	@ 4 mA
V_{OH}	Output High Voltage	2.4	V3P3S	V	@ –4 mA

33.17.2 Interface Timing Parameters and Waveforms

Table 33-35 and Figure 33-15 contain the timing specifications for the PMU_SUSCLK signal.

Table 33-35. SUS Clock (RTC Clock) Output Signal Timing Specifications

Symbol	Parameter	Min	Typ	Max	Units	Fig	Notes
	Operating frequency	–	32.768	–	kHz		
	Tolerance	100	–	100	ppm		
	Duty Cycle	40	–	60	%		
T_{SLF}	Slew Rate - Falling Edge of Clock	5	–	10	ns	33-21	1
T_{SLR}	Slew Rate - Rising Edge of Clock	5	–	10	ns	33-21	1
	Jitter	300	–	300	ps		2
T_{VAL}	PMU_SUSCLK SoC output stable after platform board de-asserts RSMRST_B	100	–	–	ms	33-15	

Notes:
1. In Figure 33-21 on page 687, V_{TL} is 20% of V3P3A and V_{TH} is 80% of V3P3A.
2. Cycle to cycle.

Figure 33-15. SUS Clock (RTC Clock) Valid Timing Diagram

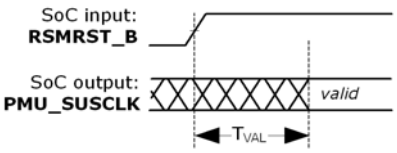


Table 33-36. Flex Clock Output Signal Timing Specifications

Symbol	Parameter	Min	Typ	Max	Units	Fig	Notes
	Operating frequency	–	25	–	MHz		25 MHz Mode
	Operating frequency	–	33	–	MHz		33 MHz Mode
	Tolerance	100	–	100	ppm		
	Duty Cycle	40	–	60	%		
T_{SLF}	Slew Rate - Falling Edge of Clock	1	–	50	ns	33-21	1
T_{SLR}	Slew Rate - Rising Edge of Clock	1	–	50	ns	33-21	1
	Jitter	300	–	300	ps		2

Notes:
1. In Figure 33-21 on page 687, V_{TL} is 20% of V3P3A and V_{TH} is 80% of V3P3A.
2. Cycle to cycle.

33.18 SoC Error-Signal Interface

33.18.1 DC Specifications

Table 33-37 contains the DC specifications for the SoC error signals. These signals are in the 3.3V Core power well of the SoC.

- NMI (Input)
- ERROR2_B (Output)
- ERROR1_B (Output)
- ERROR0_B (Output)
- IERR_B (Output)
- MCERR_B (Output)

Table 33-37. SoC Error Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	V3P3S+0.3	V	
V_{OL}	Output Low Voltage	–	0.4	V	@ 4 mA
V_{OH}	Output High Voltage	2.4	V3P3S	V	@ –4 mA

33.19 SoC Reset and Power Management Unit (PMU) Interface

33.19.1 DC Specifications

Table 33-38 and Table 33-39 contain the DC specifications for the Reset and Power Management interface signals.

This signal is in the 3.3V Core power well of the SoC. See Table 33-38.

- PMU_RESETBUTTON_B (Input)

Table 33-38. PMU_RESETBUTTON_B Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	V3P3S+0.3	V	

These signals are in the 3.3V SUS power well of the SoC. See Table 33-39

- PMU_WAKE_B (Input)
- PMU_PWRBTN_B (Input)
- PMU_PLTRST_B (Output)
- PMU_SLP_LAN_B (Output)
- PMU_SLP_S3_B (Output)
- PMU_SLP_S45_B (Output)
- PMU_SLP_DDRVTT_B (Output)
- SUSPWRDNACK (Output)
- SUS_STAT_B (Output)
- CPU_RESET_B (Output)

Table 33-39. Reset and Power Management Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.3	0.8	V	
V_{IH}	Input High Voltage	2.0	V3P3A+0.3	V	
V_{OL}	Output Low Voltage	–	0.4	V	@ 4 mA
V_{OH}	Output High Voltage	2.4	V3P3A	V	@ –4 mA

33.19.2 Interface Timing Parameters and Waveforms

The PMU interface signals, their state exchange and timing with the platform board design are in Chapter 7, “SoC Reset and Power Supply Sequences”.

33.20 SoC Real-Time Clock (RTC) Interface

The SoC requires a 32.768KHz crystal in parallel resonance mode. The DC and crystal requirements follow.

33.20.1 DC Specifications

Table 33-40 and Table 33-41 contain the DC specifications for the RTC signals. These signals are in the RTC power well of the SoC.

- RSMRST_B (Input)
- COREPWROK (Input)
- SRTCRST_B (Input)

Table 33-40. RTC Input Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.5	0.78	V	
V_{IH}	Input High Voltage	2.0	VRTC3P0+0.5	V	

- RTEST_B (Input)

Table 33-41. RTC RTEST_B Signal DC Specifications

Symbol	Parameter	Min	Max	Unit	Note
V_{IL}	Input Low Voltage	–0.5	0.78	V	
V_{IH}	Input High Voltage	2.3	VRTC3P0+0.5	V	Special V_{IH} Min for bad-battery detection

33.20.2 RTC Crystal Specifications

Table 33-42 contains the requirements for the RTC crystal.

- BRTCX1_PAD
- BRTCX2_PAD

Table 33-42. RTC Crystal Requirements

Parameter	Min	Typ	Max	Unit	Note
Frequency	–	32.768	–	kHz	1
Cut	–	AT	–	n/a	1
Loading	–	Parallel	–	n/a	1
Load Capacity (C)	–	–	20	pF	1
Drive Strength	–	–	100	µW	1
Shunt Capacity	–	0.5	1.0	pf	1
Series Resistance	–	–	80	kΩ	1
Cut Accuracy Maximum	–	±25	–	ppm	1, 2
Temperature Stability Maximum (0-50°C)	–	±20	–	ppm	1, 2
Aging Maximum	–	±5	–	ppm	1, 2

Notes:
1. These are the specifications needed to select a crystal oscillator for the RTC circuit.
2. Crystal tolerance impacts RTC time. A 10 ppm crystal is recommended for 1.7 seconds tolerance per day, RTC circuit itself contributes addition 10 ppm for a total of 20 ppm in this example.

33.20.3 Interface Timing Parameters and Waveforms

The RTC interface signals, their state exchange and timing with the platform board design are in Chapter 7, "SoC Reset and Power Supply Sequences".

33.21 SoC Thermal Management Interface

33.21.1 DC Specifications

Table 33-43 contains the DC specifications for the RTC signals. These signals are in the 1.0-Volt Core power well of the SoC.

- THERMTRIP_N (OD Output)
- PROCHOT_B (OD Output / Input)
- MEMHOT_B (Input)

Table 33-43. Thermal Signal DC Specifications

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input Low Voltage	–0.1	0.4	V	
V_{IH}	Input High Voltage	0.8	V1P0S+0.1	V	
T_{IN_SLEW}	Input slew rate required	0.10	1.05	V/ns	
V_{OL}	Output Low Voltage	–	0.1*V1P0S	V	Open Drain @ 1.5 mA
V_O	Output Voltage applied to pin	–	1.05	V	Open Drain

33.22 SoC Serial VID (SVID) Interface

33.22.1 DC Specifications

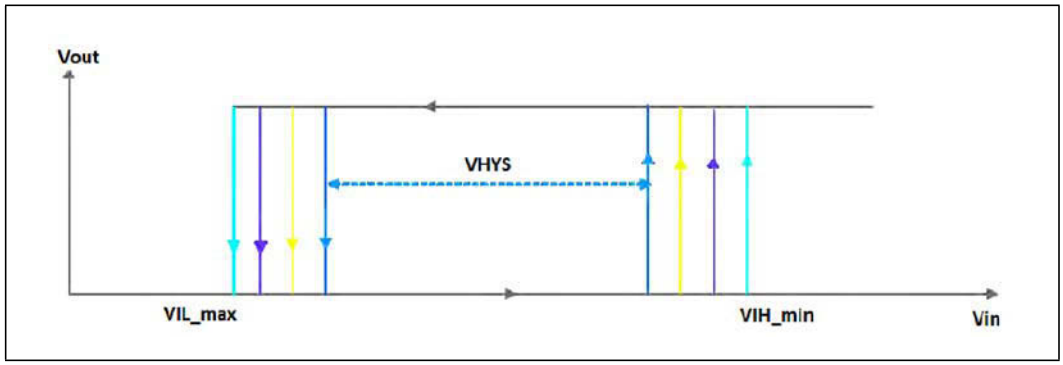
Table 33-44 and Figure 33-16 contains the DC specifications for the SVID signals. These signals are in the 1.0-Volt Core power well of the SoC.

- SVID_DATA (OD Output / Input)
- SVID_CLK (OD Output)
- SVID_ALERT_B (Input)

Table 33-44. SVID Signal DC Specifications

Symbol	Parameter	Min	Max	Units	Notes
V_{IL}	Input Low Voltage	–0.1	0.4	V	
V_{IH}	Input High Voltage	0.8	V1P0S+0.1	V	
V_{OL}	Output Low Voltage	–	0.1*V1P0S	V	Open Drain @ 1.5 mA
V_{HYS}	Hysteresis Voltage	0.05	–	V	See Figure 33-16 on page 681
V_O	Output Voltage applied to pin	–	1.05	V	Open Drain
R_{ON}	Buffer-On Resistance	10	20	Ω	Measured at 0.31 * V1P0S
I_L	Leakage Current	–100	100	µA	V_{IN} between 0V and V1P0S
C_{PAD}	Pad Capacitance	–	–4.0	pF	Die capacitance only. No package parasitic included.
V_{PIN}	Pin Capacitance	–	–5.0	pF	

Figure 33-16. SVID Hysteresis Voltage Diagram



33.22.2 Interface Timing Parameters and Waveforms

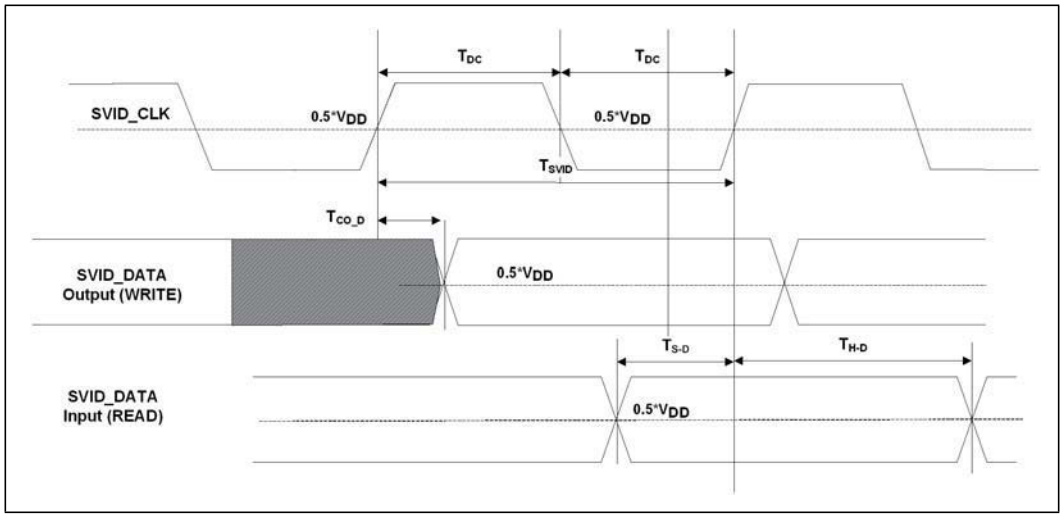
Table 33-45, and Figure 33-17 contain the timing specifications for the SVID signals.

Table 33-45. SVID Signal Timing Specifications

Symbol	Parameter	Min	Typ	Max	Units	Fig	Notes
F_{SVID}	SVID_CLK Frequency	–	25	–	MHz	33-17	
T_{DC}	SVID_CLK Duty Cycle	45	–	55	%		
T_{S_D}	SVID_DATA Input Setup Time	–2	–	–	ns	33-17	
T_{H_D}	SVID_DATA Input hold Time	9	–	–	ns	33-17	
T_{CO_D}	Rising-Edge SVID_CLK to SVID_DATA Output	0	–	5	ns	33-17	
$T_{RISE/FALL}$	Min and Max Rise/Fall Time	2	–	3	ns		1, 2

Notes:
1. Based on trace length of 0.2–4 inches, total maximum far end capacitance of 5 pF and board impedance of 25–75 Ω.
2. Measured from 30–70%.

Figure 33-17. SVID Timing Diagram



33.23 SoC JTAG and Debug Interfaces

33.23.1 DC Specifications

Table 33-46, Table 33-47 contain the DC specifications for the JTAG and Debug signals. These signals are in the 1.0-Volt SUS power well of the SoC.

- TCK (Input)
- TDI (Input)
- TMS (Input)
- TRST_B (Input)

Table 33-46. TAP and Debug Input Signal DC Specifications

Symbol	Parameter	Min	Max	Units	Notes
V_{IH}	Input High Voltage	0.85*V1P0A		V	1
V_{IL}	Input Low Voltage	–	0.35*V1P0A	V	2
Z_{pu}	Pull up Impedance	–	60	Ω	3
Z_{pd}	Pull down Impedance	–	60	Ω	3
R_{wpu}	Weak Pull Impedance	1	4	kΩ	3
R_{wpd}	Weak Pull Down Impedance	1	4	kΩ	3
$R_{wpu-40K}$	Weak Pull Up Impedance 40K	20	70	kΩ	4
$R_{wpd-40K}$	Weak Pull Down Impedance 40K	20	70	kΩ	4

Notes:
1. V_{IH} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical high value.
2. V_{IL} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical low value.
3. Measured at V1P0A ÷ 2.
4. Rwpu_40k and Rwpd_40k are only used for TRST_B.

Table 33-47. TAP and Debug Output Signal DC Specifications

Symbol	Parameter	Min	Max	Units	Notes
V_{OH}	Output High Voltage	–	1.05	V	1
V_{OL}	Output Low Voltage	0	–	V	2
R_{ON}	Buffer Resistance	25	30	Ω	3
R_{wpu}	Weak Pull Impedance	1	4	kΩ	3
R_{wpd}	Weak Pull Down Impedance	1	4	kΩ	3

Notes:
1. Minimum V_{OH} depends on the pull-down resistance on the system.
2. Maximum V_{OL} depends on the pull-up resistance on the system.
3. Measured at V1P0A ÷ 2.

Table 33-48 contains the DC specifications for the JTAG and Debug signals. These signals are in the 1.0-Volt Core power well of the SoC.

- CX_PRDY_B (Input / Output-OD)
- CX_PREQ_B (Input)

Table 33-48. TAP CX_PRDY_B and CX_PREQ_B Signal DC Specifications

Symbol	Parameter	Min	Max	Units	Notes
V_{IH}	Input High Voltage	0.8	V1POS+0.1	V	1
V_{IL}	Input Low Voltage	–0.1	V1P0S*0.4	V	2
Z_{pd}	Pull down Impedance	–	30	Ω	3
R_{wpu}	Weak Pull Impedance	1	4	kΩ	3

1. V_{IH} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical high value.
2. V_{IL} is defined as the minimum voltage level at a receiving agent that will be interpreted as a logical low value.
3. Measured at V1P0S ÷ 2.

33.23.2 Interface Timing Parameters and Waveforms

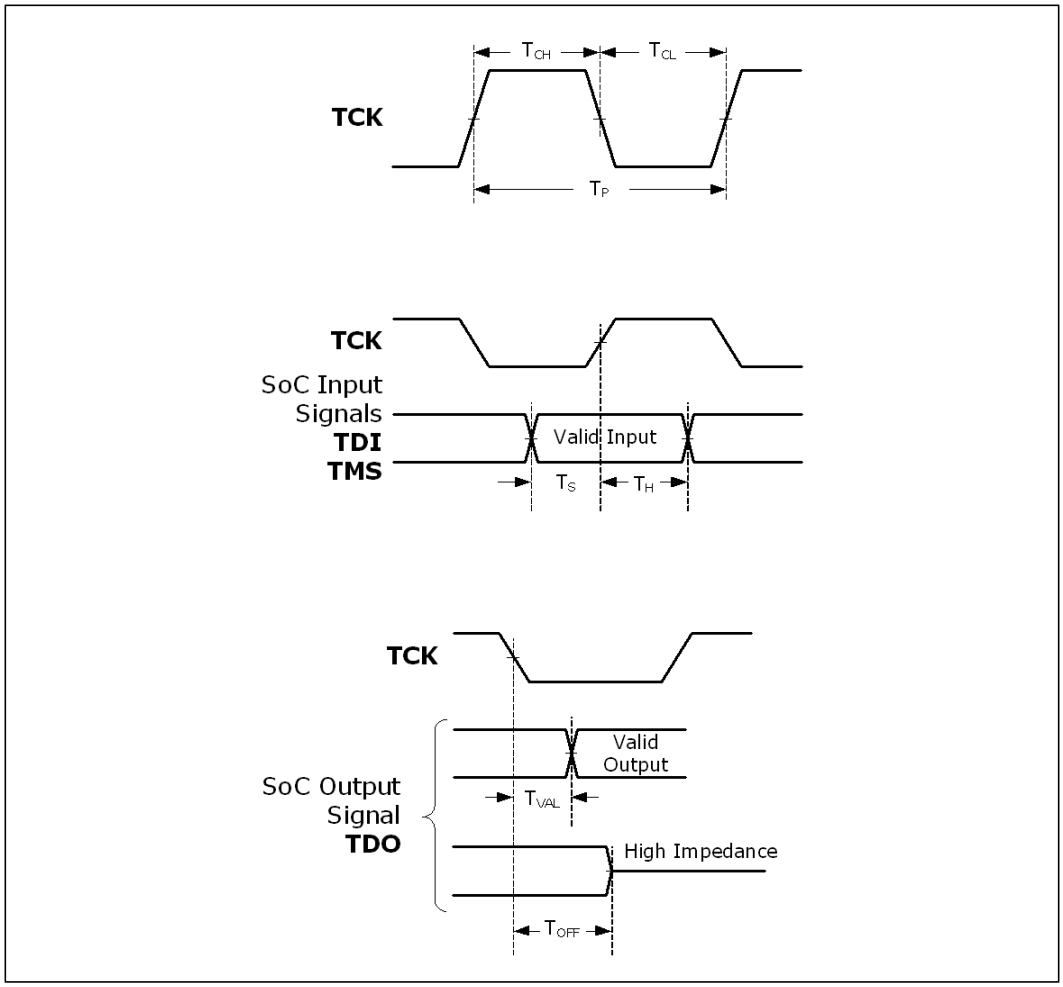
Table 33-49 and Figure 33-18 contain the timing specifications for the JTAG and Debug signals. Unless otherwise noted, all specifications in these tables apply to all SoC frequencies and a maximum platform-board JTAG-signal skew of ±500 ps. Parameters are not 100% tested and are specified by design characterization.

Table 33-49. JTAG Signal Timing Specifications

Symbol	Parameter	Min	Max	Units	Figure	Notes
T_P	TCK Period	15	–	ns	33-18	66 MHz
T_{CL}	TCK Clock Low Time	0.2 * T_P	–	ns	33-18	1
T_{CH}	TCK Clock High Time	0.2 * T_P	–	ns	33-18	1
T_S	TDI, TMS Setup Time	11	–	ns	33-18	
T_H	TDI, TMS Hold Time	5	–	ns	33-18	
T_{VAL}	TCK falling to TDO output valid	–	11	ns	33-18	
T_{OFF}	TCK falling to TDO output high impedance	–	11	ns	33-18	
T_W	TRST_B assert time	2	–	ns	33-22	2

Notes:
1. 40% of one-half of T_P (CLK Period).
2. It is recommended that TMS be asserted while TRST_B is being de-asserted.

Figure 33-18.JTAG Timing Diagram



33.24 Waveform Figures Commonly Referenced

Refer to other subsections for signal requirements and characteristics that refer to these figures.

Figure 33-19.Input and Output DC Logic Level Diagram - Single-Ended

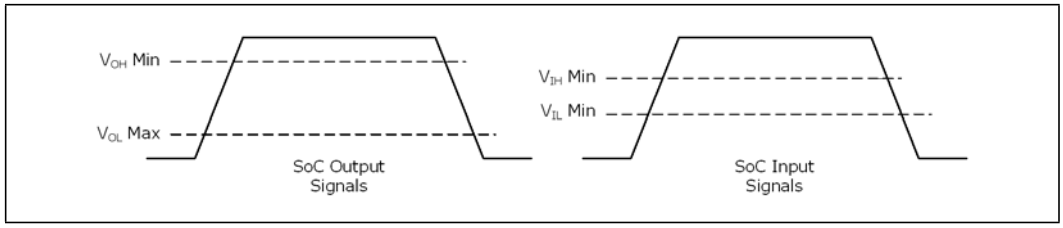


Figure 33-20.High and Low Signal Voltage Diagram - Single-Ended

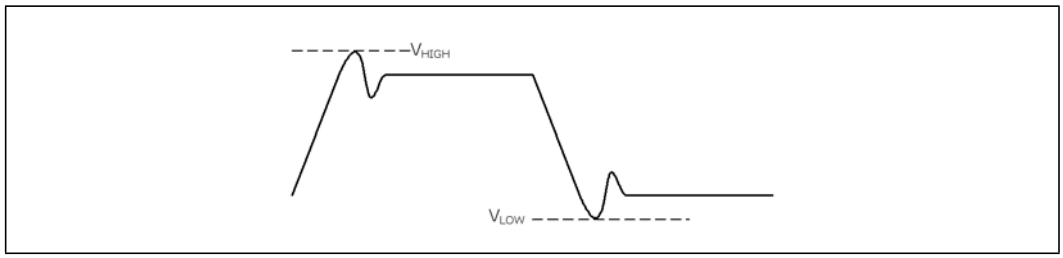


Figure 33-21.Clock Period and Slew Rate Diagram - Single-Ended

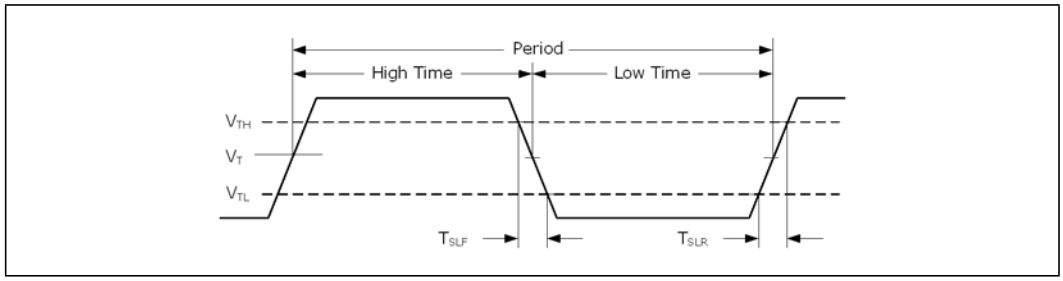
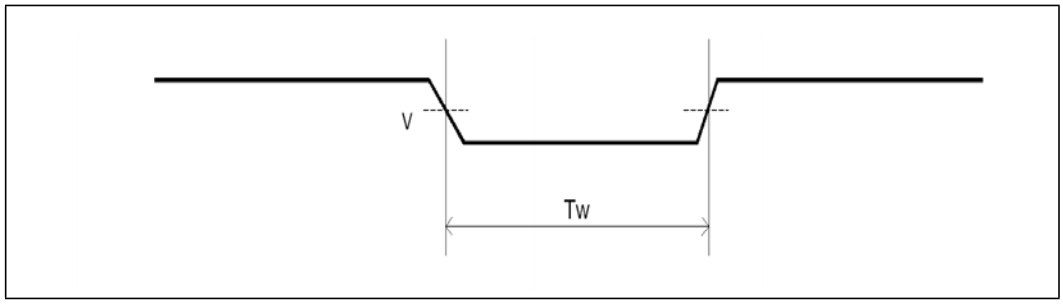


Figure 33-22.Signal Pulse Width Timing Diagram



§ §

34 Operating Conditions and Power Requirements

34.1 Absolute Maximum and Minimum Ratings

For proper functional operation, all processor electrical and thermal requirements must be satisfied. These are shown starting with Section 34.2, "Normal Operating Conditions" on page 689.

When the device is subjected to conditions outside the functional operation condition limits, but within absolute maximum and minimum ratings, neither functionality nor long-term reliability can be expected. If a device is returned to conditions within functional operation limits after having been subjected to conditions outside these limits, but within the absolute maximum and minimum ratings, the device may be functional, but with its lifetime degraded depending on exposure to conditions exceeding the functional operation condition limits.

At conditions exceeding absolute maximum and minimum ratings, neither functionality nor long-term reliability can be expected. Moreover, if a device is subjected to these conditions for any length of time, it will either not function or its reliability will be severely degraded when returned to conditions within the functional operating condition limits.

34.1.1 Component Storage Conditions Specification

This section applies to the component-level storage prior to board attach. Environmental storage condition limits define the temperature and relative humidity to which the device is exposed to while being stored in the applicable Intel shipping media trays, reels, and moisture barrier bags and boxes, and the component is not electrically connected.

34.1.1.1 Prior to Board-Attach

Table 34-1 specifies absolute maximum and minimum storage temperature and humidity limits for given time durations. Failure to adhere to the specified limits could result in physical damage to the component. If this is suspected, Intel recommends a visual inspection to determine possible physical damage to the silicon or surface components.

Table 34-1. Storage Condition Ratings - Prior to Board-Attach

Symbol	Parameter	Minimum	Maximum	Unit
$T_{absolute\ storage}$	Device storage temperature when exceeded for any length of time.	-25	125	°C
$T_{sustained\ storage\ time\ and\ temperature}$	The minimum/maximum device storage temperature for a sustained period of time.	-5	40	°C
$T_{short\ term\ storage}$	The ambient storage temperature and time for up to 72 hours.	-25	85	°C
$RH_{sustained\ storage}$	The maximum device storage relative humidity for up to 30 months.		60% at 24 °C	

Notes:
1. Specified temperatures are not to exceed values based on data collected. Exceptions for surface mount re-flow are specified by the applicable JEDEC standard. Non-adherence may affect processor reliability.
2. Component product device storage temperature qualification methods may follow JESD22-A119 (low temperature) and JESD22-A103 (high temperature) standards when applicable for volatile memory.
3. Component stress testing is conducted in conformance with JESD22-A104.
4. The JEDEC J-JSTD-020 moisture level rating and associated handling practices apply to all moisture sensitive devices removed from the moisture barrier bag.

34.1.1.2 Post Board-Attach

The storage condition limits for the component once attached to the application board are not specified.

Intel does not conduct component-level certification assessments post board-attach given the multitude of attach methods, socket types, and board types used by customers.

Provided as general guidance only, board-level Intel-branded products are specified and certified to meet the following temperature and humidity limits:

- Non-Operating Temperature Limit: -40 °C to 70 °C
- Humidity: 50% to 90%, non-condensing with a maximum wet-bulb of 28 °C

34.2 Normal Operating Conditions

34.2.1 Temperature

Table 34-2 specifies the normal operating temperature range for all product SKUs.

All supply voltage requirements, input signal requirements, and output signal characteristics are specified for the normal operating temperature range of the device. The temperature range is specified in terms of the Package Junction Temperature (Tj) which is the temperature of the die active surface. For a platform board and chassis using the device, the design must maintain an operational Tj within the specified range. Intel provides design guidance in the *Intel® Atom™ Processor C2000 Product Family for Microserver Thermal and Mechanical Specifications and Design Guidelines (TMSDG)*.

Table 34-2. Operating Temperature Range

Parameter	Symbol	Minimum	Maximum	Unit
Package Junction Temperature	Tj	0	100	°C

34.2.2 Supply Voltage and Current Requirements

Table 34-3 specifies the device voltage supply requirements for each of the voltage supply groups. These groups are described in Section 9.5, "Supply Voltage Rails" on page 156.

A voltage group name ending with an "A" signifies a supply that must always be on for all ACPI Sleep States (S0 through S5). Those ending with an "S" signifies a supply that is on only for S0 and switched off during all other states.

Table 34-3. Voltage Supply Requirements Under Normal Operating Conditions

Group	Parameter	Typical (V)	DC Tolerance	Ripple	Total Tolerance
VCC	SVID voltage for core circuitry (variable)	0.5-1.3	1.50%	1.00%	54 mV
VNN	Static SVID voltage for un-core circuitry (variable)	0.5-1.3	1.50%	1.00%	64 mV
V1P0A	Device circuitry (always-on voltage supply)	1.00	3%	1%	47 mV
V1P0S	Device circuitry (switched voltage supply)	1.00	2%	1%	47 mV
VCCSRAM (V1P1S)	Device circuitry (switched voltage supply)	1.07	3%	1%	47 mV
V1P35S	Device circuitry (switched voltage supply)	1.35	2%	See Note	40 mV
V1P8A	Device circuitry (always-on voltage supply)	1.8	3%	See Note	72 mV
V1P8S	Device circuitry (switched voltage supply)	1.8	3%	See Note	72 mV
V3P3A	Device circuitry (always-on voltage supply)	3.3	2%	See Note	132 mV
V3P3S	Device circuitry (switched voltage supply)	3.3	2%	See Note	132 mV
VDDQA	DDR3 circuitry (switched voltage supply) for standard DDR3, VDDQ = 1.5V	1.5	1.80%	1%	60 mV
	DDR3 circuitry (switched voltage supply) for low-power DDR3 (DDR3L), VDDQ = 1.35V	1.35	1.80%	1%	54 mV
VDDQB	DDR3 circuitry (switched voltage supply) for standard DDR3, VDDQ = 1.5V	1.5	1.80%	1%	60 mV
	DDR3 circuitry (switched voltage supply) for low-power DDR3 (DDR3L), VDDQ = 1.35V	1.35	1.80%	1%	54 mV
VRTC3P0	Real Time Clock (RTC) battery (always-on voltage supply including when the system is in the G3 Mechanical-Off state)	3.0	See Note	See Note	-1.0V/+0.4V

Note: For values that are not specified, platform board designs are not required to stay within any particular limit, but all other specified tolerances must be met.

Adequate supply current is required for each device voltage group to ensure proper operation under normal conditions. This current requirement varies by product SKU. The tables below show the maximum current ICC_{MAX} the device draws over the supply voltage ranges shown in Table 34-4. ICC_{MAX} is the worst-case transient current that the SoC draws from a given power supply. Platform board designers typically use this value to keep the board Voltage Regulator (VR) current below the VR Over-Current Protection (OCP) limit. The Thermal Design Current (TDC) is also shown. It is the worst-case sustained current (DC equivalent) that the SoC draws from a given power supply. Platform board designers use this value for proper VR design including VR thermal design.

Note: The TDC values shown in Table 34-4 are representative numbers. TDC values vary across parts and operating conditions. Only the ICC_{MAX} values are guaranteed.

The SoC VCC (CPU core voltage source) and all the other SoC voltage sources do not require Adaptive Voltage Positioning (AVP). All of the voltage rails have load lines equal to 0 Ω. The SoC does not have a large-core, high-current processor; therefore, the AVP provides little benefit on SoC loads less than 20A.

Table 34-4. Supply Current Required - C2750 (SKU 3)

Group	Parameter	TDC	ICC_{MAX}	Unit
VCC	SVID voltage for core circuitry (variable)	12.0	23.3	A
VNN	Static SVID voltage for un-core circuitry (variable)	2.2	2.2	A
V1P0A	Device circuitry (always-on voltage supply)	1.2	1.2	A
V1P0S	Device circuitry (switched voltage supply)	5.6	6.9	A
VCCSRAM	Device circuitry (switched voltage supply)	2.5	2.8	A
V1P35S	Device circuitry (switched voltage supply)	0.2	0.2	A
V1P8A	Device circuitry (always-on voltage supply)	0.1	0.1	A
V1P8S	Device circuitry (switched voltage supply)	0.1	0.1	A
V3P3A	Device circuitry (always-on voltage supply)	0.08	0.08	A
V3P3S	Device circuitry (switched voltage supply)	0.08	0.08	A
VDDQA	DDR3 circuitry (switched voltage supply) for Standard DDR3, VDDQ = 1.5V	2.0	2.0	A
	DDR3 circuitry (switched voltage supply) for low-power DDR3 (DDR3L), VDDQ = 1.35V	2.0	2.0	A
VDDQB	DDR3 circuitry (switched voltage supply) for Standard DDR3, VDDQ = 1.5V	2.0	2.0	A
	DDR3 circuitry (switched voltage supply) for low-power DDR3 (DDR3L), VDDQ = 1.35V	2.0	2.0	A
VRTC3P0	Real-Time Clock (RTC) (always-on voltage supply)	1.11	1.17	mA
	RTC when the system is in the G3 Mechanical-Off state when coin-battery installed	N/A	6	µA

Table 34-5. Supply Current Required - C2730 (SKU 4)

Group	Parameter	TDC	ICC_{MAX}	Unit
VCC	SVID voltage for core circuitry (variable)	6.5	12.5	A
VNN	Static SVID voltage for un-core circuitry (variable)	2.2	2.2	A
V1P0A	Device circuitry (always-on voltage supply)	1.2	1.2	A
V1P0S	Device circuitry (switched voltage supply)	4.0	5.5	A
VCCSRAM	Device circuitry (switched voltage supply)	2.5	2.8	A
V1P35S	Device circuitry (switched voltage supply)	0.2	0.2	A
V1P8A	Device circuitry (always-on voltage supply)	0.1	0.1	A
V1P8S	Device circuitry (switched voltage supply)	0.1	0.1	A
V3P3A	Device circuitry (always-on voltage supply)	0.08	0.08	A
V3P3S	Device circuitry (switched voltage supply)	0.08	0.08	A
VDDQA	DDR3 circuitry (switched voltage supply) for Standard DDR3, VDDQ = 1.5V	2.0	2.0	A
	DDR3 circuitry (switched voltage supply) for low-power DDR3 (DDR3L), VDDQ = 1.35V	2.0	2.0	A
VDDQB	DDR3 circuitry (switched voltage supply) for Standard DDR3, VDDQ = 1.5V	2.0	2.0	A
	DDR3 circuitry (switched voltage supply) for low-power DDR3 (DDR3L), VDDQ = 1.35V	2.0	2.0	A
VRTC3P0	Real-Time Clock (RTC) (always-on voltage supply)	1.11	1.17	mA
	RTC when the system is in the G3 Mechanical-Off state when coin-battery installed	N/A	6	µA

Table 34-6. Supply Current Required - C2550 (SKU 6)

Group	Parameter	TDC	ICC_{MAX}	Unit
VCC	SVID voltage for core circuitry (variable)	6.0	12.0	A
VNN	Static SVID voltage for un-core circuitry (variable)	2.2	2.2	A
V1P0A	Device circuitry (always-on voltage supply)	1.2	1.2	A
V1P0S	Device circuitry (switched voltage supply)	5.6	6.9	A
VCCSRAM	Device circuitry (switched voltage supply)	1.5	1.8	A
V1P35S	Device circuitry (switched voltage supply)	0.2	0.2	A
V1P8A	Device circuitry (always-on voltage supply)	0.1	0.1	A
V1P8S	Device circuitry (switched voltage supply)	0.1	0.1	A
V3P3A	Device circuitry (always-on voltage supply)	0.08	0.08	A
V3P3S	Device circuitry (switched voltage supply)	0.08	0.08	A
VDDQA	DDR3 circuitry (switched voltage supply) for Standard DDR3, VDDQ = 1.5V	2.0	2.0	A
	DDR3 circuitry (switched voltage supply) for low-power DDR3 (DDR3L), VDDQ = 1.35V	2.0	2.0	A
VDDQB	DDR3 circuitry (switched voltage supply) for Standard DDR3, VDDQ = 1.5V	2.0	2.0	A
	DDR3 circuitry (switched voltage supply) for low-power DDR3 (DDR3L), VDDQ = 1.35V	2.0	2.0	A
VRTC3P0	Real-Time Clock (RTC) (always-on voltage supply)	1.11	1.17	mA
	RTC when the system is in the G3 Mechanical-Off state when coin-battery installed	N/A	6	µA

Table 34-7. Supply Current Required - C2530 (SKU 7)

Group	Parameter	TDC	ICC_{MAX}	Unit
VCC	SVID voltage for core circuitry (variable)	4.0	6.0	A
VNN	Static SVID voltage for un-core circuitry (variable)	2.2	2.2	A
V1P0A	Device circuitry (always-on voltage supply)	1.2	1.2	A
V1P0S	Device circuitry (switched voltage supply)	4.0	4.0	A
VCCSRAM	Device circuitry (switched voltage supply)	1.5	1.8	A
V1P35S	Device circuitry (switched voltage supply)	0.2	0.2	A
V1P8A	Device circuitry (always-on voltage supply)	0.2	0.1	A
V1P8S	Device circuitry (switched voltage supply)	0.1	0.1	A
V3P3A	Device circuitry (always-on voltage supply)	0.08	0.08	A
V3P3S	Device circuitry (switched voltage supply)	0.80	0.08	A
VDDQA	DDR3 circuitry (switched voltage supply) for Standard DDR3, VDDQ = 1.5V	2.0	2.0	A
	DDR3 circuitry (switched voltage supply) for low-power DDR3 (DDR3L), VDDQ = 1.35V	2.0	2.0	A

Table 34-7. Supply Current Required - C2530 (SKU 7)

Group	Parameter	TDC	ICC_{MAX}	Unit
VDDQB	DDR3 circuitry (switched voltage supply) for Standard DDR3, VDDQ = 1.5V	2.0	2.0	A
	DDR3 circuitry (switched voltage supply) for low-power DDR3 (DDR3L), VDDQ = 1.35V	2.0	2.0	A
VRTC3P0	Real-Time Clock (RTC) (always-on voltage supply)	1.11	1.17	mA
	RTC when the system is in the G3 Mechanical-Off state when coin-battery installed	N/A	6	µA

Table 34-8. Supply Current Required - C2350 (SKU 8)

Group	Parameter	TDC	ICC_{MAX}	Unit
VCC	SVID voltage for core circuitry (variable)	3.0	6.0	A
VNN	Static SVID voltage for un-core circuitry (variable)	2.2	2.2	A
V1P0A	Device circuitry (always-on voltage supply)	1.2	1.2	A
V1P0S	Device circuitry (switched voltage supply)	4.0	4.0	A
VCCSRAM	Device circuitry (switched voltage supply)	1.5	1.8	A
V1P35S	Device circuitry (switched voltage supply)	0.2	0.2	A
V1P8A	Device circuitry (always-on voltage supply)	0.1	0.1	A
V1P8S	Device circuitry (switched voltage supply)	0.1	0.1	A
V3P3A	Device circuitry (always-on voltage supply)	0.08	0.08	A
V3P3S	Device circuitry (switched voltage supply)	0.08	0.08	A
VDDQ	DDR3 circuitry (switched voltage supply) for standard DDR3, VDDQ = 1.5V	3.0	3.0	A
	DDR3 circuitry (switched voltage supply) for low-power DDR3 (DDR3L), VDDQ = 1.35V	3.0	3.0	A
VRTC3P0	Real-Time Clock (RTC) (always-on voltage supply)	1.11	1.17	mA
	RTC when the system is in the G3 Mechanical-Off state when coin-battery installed	N/A	6	µA

34.2.3 Voltage Supply Pins and VR Groups

Table 34-9 shows the voltage regulator groups and the SoC voltage supply pins in each group.

Table 34-9. Voltage Supply Pins and VR Groups (Sheet 1 of 3)

Voltage Regulator Group		Voltage Supply Pin/Ball Name	Number of Pins/ Balls
CPU Core Power Source			
VCC		VCCCPUVIDSI0_1P03	42
VCC Sense (The SoC output pins to be used by VR on the platform board.)		VCCCPUVIDSI0_1P03_SENSE VSSRAMCPUSI1_1P03_SENSE	1 1
VNN Power Source			
VNN		VNN	22
DDR3 Power Source			
VDDQ (1.35V or 1.5V, Switched)	VDDQA	VCCDDR_0_1P5	24
		VCCCLKDDR_0_1P5[1]	2
	VDDQB	VCCDDR_1_1P5[1]	24
		VCCCLKDDR_1_1P5[1]	2
3.3V SUS Well Power Source			
V3P3A (3.3V, Always on)		VCCUSBSUS_3P3	2
		VCCPADXXXSUS_3P3	2
3.3V Core Well Power Source			
V3P3S (3.3V, Switched)		VCCPADXXXSI0_3P3	1
1.8V SUS Power Sources			
V1P8A (1.8V, Always on)		VCCUSBSUS_1P8	2
		VCCPADXXXSUS_1P8	2
1.8V Core Well Power Source			
V1P8S (1.8V, Switched)		VCCPADXXXSI0_1P8	2
1.35V Core Well Power Source			
V1P35S (1.35V, Switched)		VCCSFRPLLDDR_0_1P5	1
		VCCSFRPLLDDR_1_1P5	1
		VCCSFRXXXSI0_1P35	4
1.07V Core Well Power Source			
VCCSRAM (1.07V, Switched)		VCCRAMCPUSI1_1P03	7
VCCSRAM Sense (The SoC output pins to be used by VR on the platform board.)		VCCRAMCPUSI1_1P03_SENSE VSSRAMCPUSI1_1P03_SENSE	1 1

1. For SKU 8, even though DDR3 channel 1 is not used, all channel 1 VDDQ power pins must be supplied power from the VDDQ channel 0 power source.

Table 34-9. Voltage Supply Pins and VR Groups (Sheet 2 of 3)

Voltage Regulator Group	Voltage Supply Pin/Ball Name	Number of Pins/ Balls
1.0V SUS Well Power Source		
V1P0A (1.0V, Always on)	VCCA_GBE_1P0	3
	VCCAPLL_GBE_1P0	2
	VCCAREF_GBE_HVGEN	2
	VCCDUSBSUS_1P0	1
	VCCDIGXXXSUS_1P03	6
	VCCFHVSOCSI0_1P03	2
1.0V Core Well Power Source		
V1P0S (1.0V, Switched)	VCCACKDDR_0_1P0	2
	VCCACKDDR_1_1P0	2
	VCCADDR_0_1P0	5
	VCCADDR_1_1P0	5
	VCCADLLDDR_0_1P0	5
	VCCADLLDDR_1_1P0	5
	VCCPLLDDR_0_1P0	1
	VCCPLLDDR_1_1P0	1
	VCCA_PCIE_1P0	8
	VCCAPLL_PCIE_1P0	2
	VCCAREF_PCIE_HVGEN	1
	VCCA_SATA_1P0	4
	VCCA_SATA3_1P0	3
	VCCAPLL_SATA_1P0	2
	VCCAPLL_SATA3_1P0	2
	VCCAREF_SATA_HVGEN	3
	VCCDUSB_1P0	2
	VCCAUSB_1P0	1
	VCCFHVCPUSI0_MOD0_1P03	1
	VCCFHVCPUSI0_MOD1_1P03	1
	VCCFHVCPUSI0_MOD2_1P03	1
	VCCFHVCPUSI0_MOD3_1P03	1
	VCCDIGXXXSI0_1P03	4
RTC Well Power Source		
VRTC3P0	VCCRTC_3P3	1
VSS		
VSS	VSS	498
	VSSA_USB	2

Table 34-9. Voltage Supply Pins and VR Groups (Sheet 3 of 3)

Voltage Regulator Group	Voltage Supply Pin/Ball Name	Number of Pins/ Balls
Pins Reserved For Intel Use Only		
The platform board must make no connection to these pins.	VCCRAMCPUSI0GT_MOD3_1P03	1
	VCCCORE6VIDSI0GT_1P03	1
	VCCCORE7VIDSI0GT_1P03	1

1. For SKU 8, even though DDR3 channel 1 is not used, all channel 1 VDDQ power pins must be supplied power from the VDDQ channel 0 power source.

§ §

35 Component Ball-Out Listing

This chapter provides the ball-out (also called solder balls, pins, bumps) assignments for the SoC. The signal names used here are defined in Chapter 31, "Signal Names and Descriptions" of this document. The mechanical details of the ball arrangement are in Section 35.1.

Warning: The balls with the signal name NC are No-Connect balls. The platform board must not make any connections to these balls.

The ball-out assignments are first shown in alphabetical order according to the signal name (see Table 35-2), and then according to the ball-grid number assignment (see Table 35-2). Section 35.1 presents the ball map and the physical locations of the signals on the ball grid.

Table 35-1. Alphabetical Signal Listing

Signal	Ball
AA47_RSVD	AA47
AC25_RSVD	AC25
AC26_RSVD	AC26
AD53_RSVD	AD53
AG60_RSVD	AG60
AJ34_RSVD	AJ34
AL34_RSVD	AL34
AP20_RSVD	AP20
AP21_RSVD	AP21
AR51_RSVD	AR51
AR53_RSVD	AR53
AR54_RSVD	AR54
AT34_RSVD	AT34
AT51_RSVD	AT51
AU34_RSVD	AU34
BRTCX1_PAD	AJ65
BRTCX2_PAD	AJ63
BVCCRTC_EXTPAD	AD55
CLK14_IN	AM56
COREPWROK	AH60
CPU_RESET_B	Y63
CTBTRIGINOUT	AM47
CTBTRIGOUT	AL47
CX_PRDY_B	AW56
CX_PREQ_B	AY53
DDR3_0_BS[0]	BL12
DDR3_0_BS[1]	BL10
DDR3_0_BS[2]	BC8
DDR3_0_CASB	BH26
DDR3_0_CK[0]	BA21
DDR3_0_CK[1]	BF21
DDR3_0_CK[2]	BG19
DDR3_0_CK[3]	BB19
DDR3_0_CKB[0]	BC21
DDR3_0_CKB[1]	BG21
DDR3_0_CKB[2]	BH19
DDR3_0_CKB[3]	BD19
DDR3_0_CKE[0]	BG4
DDR3_0_CKE[1]	BH5
DDR3_0_CKE[2]	BG5
DDR3_0_CKE[3]	BH3
DDR3_0_CMDPU	BA26
DDR3_0_CSB[0]	BE25
DDR3_0_CSB[1]	BB25
DDR3_0_CSB[2]	BD23
DDR3_0_CSB[3]	BG26
DDR3_0_DQ[0]	AM15
DDR3_0_DQ[1]	AM14
DDR3_0_DQ[2]	AL8
DDR3_0_DQ[3]	AM8
DDR3_0_DQ[4]	AM17
DDR3_0_DQ[5]	AL15
DDR3_0_DQ[6]	AM11
DDR3_0_DQ[7]	AM9
DDR3_0_DQ[8]	AR14
DDR3_0_DQ[9]	AT13
DDR3_0_DQ[10]	AT10
DDR3_0_DQ[11]	AR11
DDR3_0_DQ[12]	AR16
DDR3_0_DQ[13]	AT14
DDR3_0_DQ[14]	AT8
DDR3_0_DQ[15]	AR10
DDR3_0_DQ[16]	AY9
DDR3_0_DQ[17]	AY11
DDR3_0_DQ[18]	AY15
DDR3_0_DQ[19]	AW17
DDR3_0_DQ[20]	AW8
DDR3_0_DQ[21]	AY8
DDR3_0_DQ[22]	AY14
DDR3_0_DQ[23]	AW15
DDR3_0_DQ[24]	AN4
DDR3_0_DQ[25]	AN2
DDR3_0_DQ[26]	AU1
DDR3_0_DQ[27]	AU3
DDR3_0_DQ[28]	AL2
DDR3_0_DQ[29]	AL4
DDR3_0_DQ[30]	AR2
DDR3_0_DQ[31]	AR4
DDR3_0_DQ[32]	BA32
DDR3_0_DQ[33]	BA30
DDR3_0_DQ[34]	BF30
DDR3_0_DQ[35]	BD30
DDR3_0_DQ[36]	BB32
DDR3_0_DQ[37]	AY30
DDR3_0_DQ[38]	BG29
DDR3_0_DQ[39]	BG30
DDR3_0_DQ[40]	BL16
DDR3_0_DQ[41]	BL17
DDR3_0_DQ[42]	BL21
DDR3_0_DQ[43]	BN21
DDR3_0_DQ[44]	BM16
DDR3_0_DQ[45]	BP16
DDR3_0_DQ[46]	BP19
DDR3_0_DQ[47]	BN19
DDR3_0_DQ[48]	BM25
DDR3_0_DQ[49]	BP25
DDR3_0_DQ[50]	BL28
DDR3_0_DQ[51]	BN28
DDR3_0_DQ[52]	BL23
DDR3_0_DQ[53]	BN23
DDR3_0_DQ[54]	BM27
DDR3_0_DQ[55]	BP28
DDR3_0_DQ[56]	BN32
DDR3_0_DQ[57]	BK32
DDR3_0_DQ[58]	BN35
DDR3_0_DQ[59]	BM36
DDR3_0_DQ[60]	BM31
DDR3_0_DQ[61]	BL32
DDR3_0_DQ[62]	BK35
DDR3_0_DQ[63]	BL35
DDR3_0_DQECC[0]	AY4
DDR3_0_DQECC[1]	AY2
DDR3_0_DQECC[2]	BD2
DDR3_0_DQECC[3]	BD4
DDR3_0_DQECC[4]	AW3
DDR3_0_DQECC[5]	AY1
DDR3_0_DQECC[6]	BC1
DDR3_0_DQECC[7]	BC3
DDR3_0_DQPU	AW21
DDR3_0_DQS[0]	AL11
DDR3_0_DQS[1]	AT7
DDR3_0_DQS[2]	AW12
DDR3_0_DQS[3]	AP3

Signal	Ball
DDR3_0_DQS[4]	BD29
DDR3_0_DQS[5]	BM18
DDR3_0_DQS[6]	BN26
DDR3_0_DQS[7]	BP34
DDR3_0_DQSB[0]	AL12
DDR3_0_DQSB[1]	AR7
DDR3_0_DQSB[2]	AW11
DDR3_0_DQSB[3]	AP1
DDR3_0_DQSB[4]	BC29
DDR3_0_DQSB[5]	BN17
DDR3_0_DQSB[6]	BL26
DDR3_0_DQSB[7]	BM34
DDR3_0_DQSBECC[0]	BB4
DDR3_0_DQSECC[0]	BB2
DDR3_0_DRAM_PWROK	BG17
DDR3_0_DRAMRSTB	BA25
DDR3_0_MA[0]	BH11
DDR3_0_MA[1]	BL8
DDR3_0_MA[2]	BG13
DDR3_0_MA[3]	BG11
DDR3_0_MA[4]	BD13
DDR3_0_MA[5]	BM7
DDR3_0_MA[6]	BD10
DDR3_0_MA[7]	BH8
DDR3_0_MA[8]	BD8
DDR3_0_MA[9]	BD15
DDR3_0_MA[10]	BN10
DDR3_0_MA[11]	BG8
DDR3_0_MA[12]	BD7
DDR3_0_MA[13]	BA23
DDR3_0_MA[14]	BC11
DDR3_0_MA[15]	BF15
DDR3_0_MON1N	BP6
DDR3_0_MON1P	BP8
DDR3_0_MON2N	BK3
DDR3_0_MON2P	BK5
DDR3_0_ODT[0]	BD26
DDR3_0_ODT[1]	BB26
DDR3_0_ODT[2]	BH25
DDR3_0_ODT[3]	BG25
DDR3_0_ODTPU	AW20

Signal	Ball
DDR3_0_RASB	BG23
DDR3_0_REFN	BM13
DDR3_0_REFP	BN12
DDR3_0_VCCA_PWROK	BE17
DDR3_0_VREF	AY29
DDR3_0_WEB	BC23
DDR3_1_BS[0]	N18
DDR3_1_BS[1]	H18
DDR3_1_BS[2]	H25
DDR3_1_CASB	H4
DDR3_1_CK[0]	C9
DDR3_1_CK[1]	G9
DDR3_1_CK[2]	H10
DDR3_1_CK[3]	J13
DDR3_1_CKB[0]	D8
DDR3_1_CKB[1]	H9
DDR3_1_CKB[2]	G10
DDR3_1_CKB[3]	L13
DDR3_1_CKE[0]	N26
DDR3_1_CKE[1]	N25
DDR3_1_CKE[2]	L25
DDR3_1_CKE[3]	L26
DDR3_1_CMDPU	T25
DDR3_1_CSB[0]	K3
DDR3_1_CSB[1]	N4
DDR3_1_CSB[2]	L7
DDR3_1_CSB[3]	K4
DDR3_1_DQ[0]	B39
DDR3_1_DQ[1]	D39
DDR3_1_DQ[2]	B35
DDR3_1_DQ[3]	D35
DDR3_1_DQ[4]	A39
DDR3_1_DQ[5]	C40
DDR3_1_DQ[6]	A36
DDR3_1_DQ[7]	C36
DDR3_1_DQ[8]	B32
DDR3_1_DQ[9]	C31
DDR3_1_DQ[10]	A27
DDR3_1_DQ[11]	C27
DDR3_1_DQ[12]	E32
DDR3_1_DQ[13]	D32

Signal	Ball
DDR3_1_DQ[14]	D28
DDR3_1_DQ[15]	B28
DDR3_1_DQ[16]	D23
DDR3_1_DQ[17]	B23
DDR3_1_DQ[18]	D19
DDR3_1_DQ[19]	B19
DDR3_1_DQ[20]	C24
DDR3_1_DQ[21]	D24
DDR3_1_DQ[22]	B21
DDR3_1_DQ[23]	D21
DDR3_1_DQ[24]	H30
DDR3_1_DQ[25]	H29
DDR3_1_DQ[26]	N32
DDR3_1_DQ[27]	L32
DDR3_1_DQ[28]	H32
DDR3_1_DQ[29]	J30
DDR3_1_DQ[30]	M30
DDR3_1_DQ[31]	P30
DDR3_1_DQ[32]	R15
DDR3_1_DQ[33]	R14
DDR3_1_DQ[34]	R9
DDR3_1_DQ[35]	R11
DDR3_1_DQ[36]	T17
DDR3_1_DQ[37]	U15
DDR3_1_DQ[38]	R8
DDR3_1_DQ[39]	U8
DDR3_1_DQ[40]	U4
DDR3_1_DQ[41]	U2
DDR3_1_DQ[42]	AB2
DDR3_1_DQ[43]	AB4
DDR3_1_DQ[44]	T3
DDR3_1_DQ[45]	T1
DDR3_1_DQ[46]	Y4
DDR3_1_DQ[47]	Y2
DDR3_1_DQ[48]	W14
DDR3_1_DQ[49]	W13
DDR3_1_DQ[50]	W10
DDR3_1_DQ[51]	Y11
DDR3_1_DQ[52]	Y16
DDR3_1_DQ[53]	Y14
DDR3_1_DQ[54]	W8

Signal	Ball
DDR3_1_DQ[55]	Y10
DDR3_1_DQ[56]	AC8
DDR3_1_DQ[57]	AD8
DDR3_1_DQ[58]	AD15
DDR3_1_DQ[59]	AC17
DDR3_1_DQ[60]	AC11
DDR3_1_DQ[61]	AC9
DDR3_1_DQ[62]	AC14
DDR3_1_DQ[63]	AC15
DDR3_1_DQECC[0]	D15
DDR3_1_DQECC[1]	C15
DDR3_1_DQECC[2]	D12
DDR3_1_DQECC[3]	E12
DDR3_1_DQECC[4]	D17
DDR3_1_DQECC[5]	B17
DDR3_1_DQECC[6]	C13
DDR3_1_DQECC[7]	B12
DDR3_1_DQPU	T26
DDR3_1_DQS[0]	D37
DDR3_1_DQS[1]	D30
DDR3_1_DQS[2]	A21
DDR3_1_DQS[3]	M29
DDR3_1_DQS[4]	U11
DDR3_1_DQS[5]	W1
DDR3_1_DQS[6]	Y7
DDR3_1_DQS[7]	AD12
DDR3_1_DQSB[0]	B37
DDR3_1_DQSB[1]	B30
DDR3_1_DQSB[2]	C22
DDR3_1_DQSB[3]	L29
DDR3_1_DQSB[4]	U12
DDR3_1_DQSB[5]	W3
DDR3_1_DQSB[6]	W7
DDR3_1_DQSB[7]	AD11
DDR3_1_DQSBECC[0]	D14
DDR3_1_DQSECC[0]	B14
DDR3_1_DRAM_PWROK	L17
DDR3_1_DRAMRSTB	N11
DDR3_1_MA[0]	H17
DDR3_1_MA[1]	G17
DDR3_1_MA[2]	G18

Signal	Ball
DDR3_1_MA[3]	P21
DDR3_1_MA[4]	M21
DDR3_1_MA[5]	H22
DDR3_1_MA[6]	J21
DDR3_1_MA[7]	J22
DDR3_1_MA[8]	H21
DDR3_1_MA[9]	M22
DDR3_1_MA[10]	L18
DDR3_1_MA[11]	P22
DDR3_1_MA[12]	G25
DDR3_1_MA[13]	G3
DDR3_1_MA[14]	G26
DDR3_1_MA[15]	H26
DDR3_1_MON1N	C5
DDR3_1_MON1P	E5
DDR3_1_MON2N	A6
DDR3_1_MON2P	A8
DDR3_1_ODT[0]	L2
DDR3_1_ODT[1]	N2
DDR3_1_ODT[2]	N8
DDR3_1_ODT[3]	L4
DDR3_1_ODTPU	T32
DDR3_1_RASB	L8
DDR3_1_REFN	L14
DDR3_1_REFP	J14
DDR3_1_VCCA_PWROK	N17
DDR3_1_VREF	R29
DDR3_1_WEB	N7
DFX_PORT_CLK0	AV63
DFX_PORT_CLK1	AV65
DFX_PORT0	BC64
DFX_PORT1	AY58
DFX_PORT2	AW58
DFX_PORT3	AR57
DFX_PORT4	AT54
DFX_PORT5	AY56
DFX_PORT6	AY63
DFX_PORT7	AW62
DFX_PORT8	AY59
DFX_PORT9	AT60
DFX_PORT10	BB63

Signal	Ball
DFX_PORT11	AT59
DFX_PORT12	AW64
DFX_PORT13	AW66
DFX_PORT14	AY65
DFX_PORT15	AR59
ERROR0_B	AL65
ERROR1_B	AL62
ERROR2_B	AL63
FLEX_CLK_SE0	AH59
FLEX_CLK_SE1	AG56
GBE_EE_CS_N	R59
GBE_EE_DI	W51
GBE_EE_DO	W60
GBE_EE_SK	T50
GBE_LED0	P46
GBE_LED1	W50
GBE_LED2	P48
GBE_LED3	R58
GBE_MDIO0_I2C_CLK	W56
GBE_MDIO0_I2C_DATA	W59
GBE_MDIO1_I2C_CLK	Y54
GBE_MDIO1_I2C_DATA	Y53
GBE_OBSN	H50
GBE_OBSP	G50
GBE_REFCLKN	D50
GBE_REFCLKP	B50
GBE_RXN[0]	K48
GBE_RXN[1]	L46
GBE_RXN[2]	L44
GBE_RXN[3]	H42
GBE_RXP[0]	H48
GBE_RXP[1]	J46
GBE_RXP[2]	J44
GBE_RXP[3]	G42
GBE_SDP0_0	T58
GBE_SDP0_1	T48
GBE_SMBALRT_N	T55
GBE_SMBCLK	P50
GBE_SMBD	T59
GBE_TXN[0]	B48

Signal	Ball
GBE_TXN[1]	B46
GBE_TXN[2]	B44
GBE_TXN[3]	A42
GBE_TXP[0]	D48
GBE_TXP[1]	D46
GBE_TXP[2]	D44
GBE_TXP[3]	C42
GPIO_SUS0	V66
GPIO_SUS1	W54
GPIO_SUS2	T53
HPLL_REFN	J37
HPLL_REFP	L37
IERR_B	AM52
ILB_SERIRQ	AT50
J38_RSVD	J38
L38_RSVD	L38
LPC_AD0	AG54
LPC_AD1	AM53
LPC_AD2	AL53
LPC_AD3	AG59
LPC_CLKOUT0	AG51
LPC_CLKOUT1	AM49
LPC_CLKRUNB	AH48
LPC_FRAMEB	AH56
MCERR_B	AL52
MEMHOT_B	AL46
NCSI_ARB_OUT	Y59
NCSI_RXD1	V63
NMI	AL56
P38_RSVD	P38
PCIE_OBSN	BA38
PCIE_OBSP	BC38
PCIE_REFCLKN	BD49
PCIE_REFCLKP	BB49
PCIE_RXN[0]	BE64
PCIE_RXN[1]	BH64
PCIE_RXN[2]	BK62
PCIE_RXN[3]	BH58
PCIE_RXN[4]	BH57
PCIE_RXN[5]	BF54
PCIE_RXN[6]	BG53

Signal	Ball
PCIE_RXN[7]	BG50
PCIE_RXN[8]	BH49
PCIE_RXN[9]	BF46
PCIE_RXN[10]	BG45
PCIE_RXN[11]	BH42
PCIE_RXN[12]	BG41
PCIE_RXN[13]	BG38
PCIE_RXN[14]	BF37
PCIE_RXN[15]	BG34
PCIE_RXP[0]	BE63
PCIE_RXP[1]	BJ63
PCIE_RXP[2]	BJ62
PCIE_RXP[3]	BG58
PCIE_RXP[4]	BG57
PCIE_RXP[5]	BD54
PCIE_RXP[6]	BF53
PCIE_RXP[7]	BE50
PCIE_RXP[8]	BG49
PCIE_RXP[9]	BD46
PCIE_RXP[10]	BF45
PCIE_RXP[11]	BG42
PCIE_RXP[12]	BE41
PCIE_RXP[13]	BF38
PCIE_RXP[14]	BD37
PCIE_RXP[15]	BE34
PCIE_TXN[0]	BL61
PCIE_TXN[1]	BM58
PCIE_TXN[2]	BN57
PCIE_TXN[3]	BN55
PCIE_TXN[4]	BK54
PCIE_TXN[5]	BL53
PCIE_TXN[6]	BM52
PCIE_TXN[7]	BL50
PCIE_TXN[8]	BM49
PCIE_TXN[9]	BL48
PCIE_TXN[10]	BL46
PCIE_TXN[11]	BL44
PCIE_TXN[12]	BM43
PCIE_TXN[13]	BN41
PCIE_TXN[14]	BK40
PCIE_TXN[15]	BL39

Signal	Ball
PCIE_TXP[0]	BK60
PCIE_TXP[1]	BL59
PCIE_TXP[2]	BL57
PCIE_TXP[3]	BL55
PCIE_TXP[4]	BM54
PCIE_TXP[5]	BN53
PCIE_TXP[6]	BL52
PCIE_TXP[7]	BN50
PCIE_TXP[8]	BP49
PCIE_TXP[9]	BN48
PCIE_TXP[10]	BN46
PCIE_TXP[11]	BN44
PCIE_TXP[12]	BP43
PCIE_TXP[13]	BL41
PCIE_TXP[14]	BM40
PCIE_TXP[15]	BN39
AB63_RSVD	AB63
PMU_PLTRST_B	AE62
PMU_PWRBTN_B	AC49
PMU_RESETBUTTON_B	AM58
PMU_SLP_DDRVTT_B	AC52
PMU_SLP_LAN_B	Y50
PMU_SLP_S3_B	AF65
PMU_SLP_S45_B	V64
PMU_SUSCLK	AD58
PMU_WAKE_B	AD66
PROCHOT_B	BB59
R37_RSVD	R37
RCOMP_CORE_LVT	AG46
RSMRST_B	AC47
RTEST_B	AD50
SATA_GP0	AT63
SATA_LEDN	AL49
SATA_OBSN	L52
SATA_OBSP	J52
SATA_REFCLKN	L54
SATA_REFCLKP	J54
SATA_RXN[0]	H63
SATA_RXN[1]	G64
SATA_RXN[2]	C62
SATA_RXN[3]	C60

Signal	Ball
SATA_RXP[0]	H62
SATA_RXP[1]	G62
SATA_RXP[2]	E62
SATA_RXP[3]	D60
SATA_TXN[0]	D57
SATA_TXN[1]	E57
SATA_TXN[2]	A54
SATA_TXN[3]	B53
SATA_TXP[0]	B57
SATA_TXP[1]	E58
SATA_TXP[2]	C54
SATA_TXP[3]	D53
SATA3_GP0	AH51
SATA3_LEDN	AH54
SATA3_OBSN	R53
SATA3_OBSP	R55
SATA3_REFCLKN	L59
SATA3_REFCLKP	L57
SATA3_RXN[0]	M64
SATA3_RXN[1]	L65
SATA3_RXP[0]	M62
SATA3_RXP[1]	L63
SATA3_TXN[0]	T64
SATA3_TXN[1]	R65
SATA3_TXP[0]	R66
SATA3_TXP[1]	R63
SMB_CLK0	AN62
SMB_CLK1	AR63
SMB_CLK2	AR65
SMB_DATA0	AP62
SMB_DATA1	AN63
SMB_DATA2	AN65
SMBALRT_N0	AL58
SPI_CLK	AF46
SPI_CS0_B	Y65
SPI_CS1_B	AC58
SPI_MISO	AD59
SPI_MOSI	AB62
SRTCRST_B	AD49
SUS_STAT_B	AB65
SUSPWRDNACK	Y57

Signal	Ball
SVID_ALERT_B	BC62
SVID_CLK	BB60
SVID_DATA	AW53
TCK	AD65
TDI	AC56
TDO	AF63
THERMTRIP_N	AT56
TMS	AC53
TRST_B	AA46
UART1_RXD	AG50
UART1_TXD	AH50
USB_DN[0]	BA46
USB_DN[1]	BC45
USB_DN[2]	BB42
USB_DN[3]	BA41
USB_DP[0]	AY46
USB_DP[1]	BA45
USB_DP[2]	BD42
USB_DP[3]	BB41
USB_OBSP	AR48
USB_OC0_B	AD63
USB_RCOMPI	AT46
USB_RCOMPO	AT48
USB_REFCLKN	BA50
USB_REFCLKP	BB50
VCCA_GBE_1P0	T37
VCCA_GBE_1P0	V38
VCCA_GBE_1P0	W38
VCCA_PCIE_1P0	AT36
VCCA_PCIE_1P0	AT38
VCCA_PCIE_1P0	AU36
VCCA_PCIE_1P0	AU38
VCCA_PCIE_1P0	AU39
VCCA_PCIE_1P0	AW36
VCCA_PCIE_1P0	AW38
VCCA_PCIE_1P0	AW39
VCCA_SATA_1P0	K41
VCCA_SATA_1P0	L41
VCCA_SATA_1P0	N41
VCCA_SATA_1P0	P41
VCCA_SATA3_1P0	L42

Signal	Ball
VCCA_SATA3_1P0	N42
VCCA_SATA3_1P0	P42
VCCACKDDR_0_1P0	AT32
VCCACKDDR_0_1P0	AU32
VCCACKDDR_1_1P0	V28
VCCACKDDR_1_1P0	W28
VCCADDR_0_1P0	AT21
VCCADDR_0_1P0	AT23
VCCADDR_0_1P0	AT25
VCCADDR_0_1P0	AT26
VCCADDR_0_1P0	AT28
VCCADDR_1_1P0	W21
VCCADDR_1_1P0	W23
VCCADDR_1_1P0	W25
VCCADDR_1_1P0	W26
VCCADDR_1_1P0	AA21
VCCADLLDDR_0_1P0	AU21
VCCADLLDDR_0_1P0	AU23
VCCADLLDDR_0_1P0	AU25
VCCADLLDDR_0_1P0	AU26
VCCADLLDDR_0_1P0	AU28
VCCADLLDDR_1_1P0	V20
VCCADLLDDR_1_1P0	V21
VCCADLLDDR_1_1P0	V23
VCCADLLDDR_1_1P0	V25
VCCADLLDDR_1_1P0	V26
VCCAPLL_GBE_1P0	V41
VCCAPLL_GBE_1P0	W41
VCCAPLL_PCIE_1P0	AY37
VCCAPLL_PCIE_1P0	BA37
VCCAPLL_SATA_1P0	R46
VCCAPLL_SATA_1P0	T45
VCCAPLL_SATA3_1P0	R44
VCCAPLL_SATA3_1P0	T44
VCCAREF_GBE_HVGEN	T39
VCCAREF_GBE_HVGEN	V39
VCCAREF_PCIE_HVGEN	AT39
VCCAREF_SATA_HVGEN	V46
VCCAREF_SATA_HVGEN	W44

Signal	Ball
VCCAREF_SATA_HVGEN	W46
VCCAUSB_1P0	AT42
VCCCLKDDR_0_1P5	AT29
VCCCLKDDR_0_1P5	AU29
VCCCLKDDR_1_1P5	V29
VCCCLKDDR_1_1P5	W29
VCCCORE6VIDSI0GT_1P03	AL32
VCCCORE7VIDSI0GT_1P03	AL31
VCCCPUVIDSI0_1P03	AD21
VCCCPUVIDSI0_1P03	AD23
VCCCPUVIDSI0_1P03	AD25
VCCCPUVIDSI0_1P03	AD26
VCCCPUVIDSI0_1P03	AE1
VCCCPUVIDSI0_1P03	AE5
VCCCPUVIDSI0_1P03	AF4
VCCCPUVIDSI0_1P03	AF20
VCCCPUVIDSI0_1P03	AF21
VCCCPUVIDSI0_1P03	AF23
VCCCPUVIDSI0_1P03	AF25
VCCCPUVIDSI0_1P03	AF26
VCCCPUVIDSI0_1P03	AG7
VCCCPUVIDSI0_1P03	AG8
VCCCPUVIDSI0_1P03	AG11
VCCCPUVIDSI0_1P03	AG13
VCCCPUVIDSI0_1P03	AG14
VCCCPUVIDSI0_1P03	AG17
VCCCPUVIDSI0_1P03	AG19
VCCCPUVIDSI0_1P03	AG21
VCCCPUVIDSI0_1P03	AH1
VCCCPUVIDSI0_1P03	AH3
VCCCPUVIDSI0_1P03	AH4
VCCCPUVIDSI0_1P03	AH7
VCCCPUVIDSI0_1P03	AH8
VCCCPUVIDSI0_1P03	AH10
VCCCPUVIDSI0_1P03	AH13
VCCCPUVIDSI0_1P03	AH14
VCCCPUVIDSI0_1P03	AH16
VCCCPUVIDSI0_1P03	AH19
VCCCPUVIDSI0_1P03	AJ2
VCCCPUVIDSI0_1P03	AJ4

Signal	Ball
VCCCPUVIDSI0_1P03	AJ5
VCCCPUVIDSI0_1P03	AJ21
VCCCPUVIDSI0_1P03	AJ23
VCCCPUVIDSI0_1P03	AJ25
VCCCPUVIDSI0_1P03	AJ26
VCCCPUVIDSI0_1P03	AK20
VCCCPUVIDSI0_1P03	AL21
VCCCPUVIDSI0_1P03	AL23
VCCCPUVIDSI0_1P03	AL25
VCCCPUVIDSI0_1P03	AL26
VCCCPUVIDSI0_1P03_SENSE	AE3
VCCDDR_0_1P5	AY19
VCCDDR_0_1P5	AY21
VCCDDR_0_1P5	AY23
VCCDDR_0_1P5	BC7
VCCDDR_0_1P5	BC10
VCCDDR_0_1P5	BC13
VCCDDR_0_1P5	BC15
VCCDDR_0_1P5	BD17
VCCDDR_0_1P5	BD21
VCCDDR_0_1P5	BD25
VCCDDR_0_1P5	BE19
VCCDDR_0_1P5	BE26
VCCDDR_0_1P5	BF13
VCCDDR_0_1P5	BF23
VCCDDR_0_1P5	BG9
VCCDDR_0_1P5	BG15
VCCDDR_0_1P5	BH17
VCCDDR_0_1P5	BJ4
VCCDDR_0_1P5	BK6
VCCDDR_0_1P5	BK9
VCCDDR_0_1P5	BK12
VCCDDR_0_1P5	BK13
VCCDDR_0_1P5	BL7
VCCDDR_0_1P5	BM9
VCCDDR_1_1P5	E7
VCCDDR_1_1P5	E9
VCCDDR_1_1P5	G5
VCCDDR_1_1P5	G7
VCCDDR_1_1P5	H13
VCCDDR_1_1P5	H14

Signal	Ball
VCCDDR_1_1P5	J5
VCCDDR_1_1P5	K10
VCCDDR_1_1P5	K17
VCCDDR_1_1P5	K18
VCCDDR_1_1P5	K25
VCCDDR_1_1P5	K26
VCCDDR_1_1P5	L5
VCCDDR_1_1P5	L21
VCCDDR_1_1P5	L22
VCCDDR_1_1P5	M3
VCCDDR_1_1P5	N13
VCCDDR_1_1P5	N14
VCCDDR_1_1P5	P17
VCCDDR_1_1P5	P18
VCCDDR_1_1P5	P25
VCCDDR_1_1P5	P26
VCCDDR_1_1P5	R21
VCCDDR_1_1P5	R22
VCCDIGXXXSI0_1P03	AD38
VCCDIGXXXSI0_1P03	AF38
VCCDIGXXXSI0_1P03	AG38
VCCDIGXXXSI0_1P03	AJ38
VCCDIGXXXSUS_1P03	AA38
VCCDIGXXXSUS_1P03	AA39
VCCDIGXXXSUS_1P03	AA41
VCCDIGXXXSUS_1P03	AC38
VCCDIGXXXSUS_1P03	AC39
VCCDIGXXXSUS_1P03	AC41
VCCDUSB_1P0	AU41
VCCDUSB_1P0	AW41
VCCDUSBSUS_1P0	AT41
VCCFHVCPUSI0_MOD0_1P03	AD28
VCCFHVCPUSI0_MOD1_1P03	AL28
VCCFHVCPUSI0_MOD2_1P03	AD29
VCCFHVCPUSI0_MOD3_1P03	AJ29
VCCFHVSOCSI0_1P03	AC42
VCCFHVSOCSI0_1P03	AC44
VCCPADXXXSI0_1P8	AJ41
VCCPADXXXSI0_1P8	AL41

Signal	Ball
VCCPADXXXSI0_3P3	AJ42
VCCPADXXXSUS_1P8	AF41
VCCPADXXXSUS_1P8	AF42
VCCPADXXXSUS_3P3	AD42
VCCPADXXXSUS_3P3	AD44
VCCPLLDDR_0_1P0	AU31
VCCPLLDDR_1_1P0	W31
VCCRAMCPUSI1_1P03	AD31
VCCRAMCPUSI1_1P03	AD32
VCCRAMCPUSI1_1P03	AF29
VCCRAMCPUSI1_1P03	AF31
VCCRAMCPUSI1_1P03	AG29
VCCRAMCPUSI1_1P03	AG31
VCCRAMCPUSI1_1P03	AG32
VCCRAMCPUSI1_1P03_SENSE	AF32
VCCRAMCPUSI0GT_MOD3_1P03	AJ31
VCCRTC_3P3	AG42
VCCSFRPLLDDR_0_1P5	AT31
VCCSFRPLLDDR_1_1P5	V31
VCCSFRXXXSI0_1P35	V34
VCCSFRXXXSI0_1P35	W34
VCCSFRXXXSI0_1P35	AA32
VCCSFRXXXSI0_1P35	AA34
VCCUSBSUS_1P8	AU46
VCCUSBSUS_1P8	AU47
VCCUSBSUS_3P3	AU42
VCCUSBSUS_3P3	AW42
VNN	AD36
VNN	AD39
VNN	AF36
VNN	AF39
VNN	AG36
VNN	AG39
VNN	AJ36
VNN	AJ39
VNN	AL36
VNN	AL39
VNN	AM25
VNN	AM26

Signal	Ball
VNN	AM28
VNN	AM29
VNN	AM31
VNN	AM32
VNN	AM34
VNN	AM36
VNN	AM38
VNN	AM39
VNN	AM41
VNN	AM42
VSS	A5
VSS	A9
VSS	A12
VSS	A15
VSS	A18
VSS	A24
VSS	A30
VSS	A33
VSS	A45
VSS	A48
VSS	A51
VSS	A57
VSS	A59
VSS	A61
VSS	A62
VSS	A64
VSS	A66
VSS	B10
VSS	B26
VSS	B41
VSS	B55
VSS	C3
VSS	C6
VSS	C18
VSS	C33
VSS	C45
VSS	C49
VSS	C51
VSS	C58
VSS	C64
VSS	C66

Signal	Ball
VSS	D6
VSS	D10
VSS	D26
VSS	D33
VSS	D41
VSS	D42
VSS	D51
VSS	D55
VSS	D59
VSS	E1
VSS	E3
VSS	E10
VSS	E13
VSS	E14
VSS	E16
VSS	E18
VSS	E19
VSS	E21
VSS	E22
VSS	E23
VSS	E25
VSS	E27
VSS	E28
VSS	E30
VSS	E31
VSS	E34
VSS	E36
VSS	E37
VSS	E39
VSS	E40
VSS	E41
VSS	E43
VSS	E45
VSS	E46
VSS	E48
VSS	E49
VSS	E50
VSS	E52
VSS	E54
VSS	E55
VSS	E59

Signal	Ball
VSS	E61
VSS	E64
VSS	E66
VSS	F1
VSS	F4
VSS	F5
VSS	F63
VSS	F66
VSS	G32
VSS	G34
VSS	G41
VSS	G48
VSS	G56
VSS	G59
VSS	H1
VSS	H34
VSS	H37
VSS	H38
VSS	H41
VSS	H44
VSS	H46
VSS	H52
VSS	H54
VSS	H56
VSS	H58
VSS	H59
VSS	H66
VSS	J29
VSS	J63
VSS	J64
VSS	J66
VSS	K1
VSS	K32
VSS	K34
VSS	K42
VSS	K50
VSS	K62
VSS	L30
VSS	L34
VSS	L48
VSS	L50

Signal	Ball
VSS	L60
VSS	M5
VSS	M37
VSS	M38
VSS	M44
VSS	M46
VSS	M52
VSS	M54
VSS	M56
VSS	M57
VSS	M59
VSS	M60
VSS	M66
VSS	N1
VSS	N5
VSS	N10
VSS	N34
VSS	N48
VSS	N50
VSS	N62
VSS	N63
VSS	N65
VSS	P29
VSS	P32
VSS	P34
VSS	P37
VSS	P44
VSS	R2
VSS	R4
VSS	R5
VSS	R12
VSS	R30
VSS	R38
VSS	R52
VSS	R56
VSS	R62
VSS	T5
VSS	T18
VSS	T20
VSS	T21
VSS	T23

Signal	Ball
VSS	T28
VSS	T29
VSS	T31
VSS	T34
VSS	T36
VSS	T41
VSS	T42
VSS	T47
VSS	T52
VSS	T56
VSS	T62
VSS	U9
VSS	U14
VSS	U62
VSS	U63
VSS	U65
VSS	V5
VSS	V32
VSS	V36
VSS	V42
VSS	V44
VSS	W4
VSS	W11
VSS	W16
VSS	W17
VSS	W19
VSS	W32
VSS	W36
VSS	W39
VSS	W42
VSS	W48
VSS	W53
VSS	W57
VSS	W62
VSS	Y5
VSS	Y8
VSS	Y13
VSS	Y17
VSS	Y19
VSS	Y48
VSS	Y51

Signal	Ball
VSS	Y56
VSS	Y60
VSS	AA3
VSS	AA5
VSS	AA23
VSS	AA25
VSS	AA26
VSS	AA28
VSS	AA29
VSS	AA31
VSS	AA36
VSS	AA42
VSS	AA44
VSS	AA62
VSS	AA64
VSS	AA66
VSS	AB1
VSS	AB5
VSS	AB20
VSS	AC12
VSS	AC18
VSS	AC20
VSS	AC21
VSS	AC23
VSS	AC28
VSS	AC29
VSS	AC31
VSS	AC32
VSS	AC34
VSS	AC36
VSS	AC46
VSS	AC50
VSS	AC55
VSS	AC59
VSS	AD2
VSS	AD4
VSS	AD5
VSS	AD9
VSS	AD14
VSS	AD17
VSS	AD18

Signal	Ball
VSS	AD20
VSS	AD34
VSS	AD41
VSS	AD46
VSS	AD47
VSS	AD52
VSS	AD56
VSS	AD62
VSS	AE47
VSS	AE64
VSS	AF28
VSS	AF44
VSS	AF62
VSS	AG5
VSS	AG10
VSS	AG16
VSS	AG23
VSS	AG25
VSS	AG26
VSS	AG28
VSS	AG34
VSS	AG41
VSS	AG44
VSS	AG48
VSS	AG53
VSS	AG57
VSS	AG63
VSS	AG64
VSS	AG66
VSS	AH11
VSS	AH17
VSS	AH53
VSS	AH57
VSS	AH62
VSS	AJ28
VSS	AJ32
VSS	AJ44
VSS	AJ46
VSS	AJ47
VSS	AK3
VSS	AK5

Signal	Ball
VSS	AK62
VSS	AK64
VSS	AK66
VSS	AL1
VSS	AL5
VSS	AL9
VSS	AL14
VSS	AL17
VSS	AL18
VSS	AL20
VSS	AL29
VSS	AL38
VSS	AL42
VSS	AL44
VSS	AL50
VSS	AL55
VSS	AL59
VSS	AM12
VSS	AM18
VSS	AM20
VSS	AM21
VSS	AM23
VSS	AM44
VSS	AM46
VSS	AM50
VSS	AM55
VSS	AM59
VSS	AN5
VSS	AN47
VSS	AN66
VSS	AP5
VSS	AP23
VSS	AP25
VSS	AP26
VSS	AP28
VSS	AP29
VSS	AP31
VSS	AP32
VSS	AP34
VSS	AP36
VSS	AP38

Signal	Ball
VSS	AP39
VSS	AP41
VSS	AP42
VSS	AP44
VSS	AP46
VSS	AP64
VSS	AR8
VSS	AR13
VSS	AR17
VSS	AR19
VSS	AR50
VSS	AR56
VSS	AR60
VSS	AR62
VSS	AT5
VSS	AT11
VSS	AT16
VSS	AT17
VSS	AT19
VSS	AT44
VSS	AT53
VSS	AT57
VSS	AT64
VSS	AT66
VSS	AU4
VSS	AU62
VSS	AV2
VSS	AV4
VSS	AV5
VSS	AW5
VSS	AW9
VSS	AW14
VSS	AW19
VSS	AW23
VSS	AW25
VSS	AW26
VSS	AW28
VSS	AW30
VSS	AW31
VSS	AW32
VSS	AW34

Signal	Ball
VSS	AW46
VSS	AW47
VSS	AW49
VSS	AW50
VSS	AW52
VSS	AW55
VSS	AW59
VSS	AY5
VSS	AY12
VSS	AY38
VSS	AY45
VSS	AY52
VSS	AY55
VSS	AY62
VSS	BA17
VSS	BA29
VSS	BA34
VSS	BA42
VSS	BA49
VSS	BB5
VSS	BB17
VSS	BB34
VSS	BB53
VSS	BB54
VSS	BB56
VSS	BB57
VSS	BB62
VSS	BB65
VSS	BB66
VSS	BC5
VSS	BC30
VSS	BC37
VSS	BC46
VSS	BD32
VSS	BD34
VSS	BD38
VSS	BD41
VSS	BD45
VSS	BD50
VSS	BD53
VSS	BD59

Signal	Ball
VSS	BD60
VSS	BD62
VSS	BD63
VSS	BD65
VSS	BE5
VSS	BE32
VSS	BE42
VSS	BE49
VSS	BE57
VSS	BE66
VSS	BF1
VSS	BF3
VSS	BF4
VSS	BF29
VSS	BF62
VSS	BG1
VSS	BG32
VSS	BG37
VSS	BG46
VSS	BG54
VSS	BG63
VSS	BG66
VSS	BH32
VSS	BH34
VSS	BH41
VSS	BH50
VSS	BH60
VSS	BH62
VSS	BJ1
VSS	BJ66
VSS	BK1
VSS	BK8
VSS	BK10
VSS	BK15
VSS	BK17
VSS	BK18
VSS	BK19
VSS	BK21
VSS	BK22
VSS	BK24
VSS	BK26

Signal	Ball
VSS	BK27
VSS	BK28
VSS	BK30
VSS	BK31
VSS	BK36
VSS	BK37
VSS	BK39
VSS	BK42
VSS	BK44
VSS	BK45
VSS	BK46
VSS	BK48
VSS	BK49
VSS	BK51
VSS	BK53
VSS	BK56
VSS	BK57
VSS	BK58
VSS	BK64
VSS	BK66
VSS	BL14
VSS	BL19
VSS	BL25
VSS	BL30
VSS	BL34
VSS	BL37
VSS	BL43
VSS	BM1
VSS	BM3
VSS	BM5
VSS	BM22
VSS	BM45
VSS	BM61
VSS	BM62
VSS	BM64
VSS	BM66
VSS	BN14
VSS	BN30
VSS	BN37
VSS	BP1
VSS	BP3

Signal	Ball
VSS	BP5
VSS	BP10
VSS	BP13
VSS	BP22
VSS	BP31
VSS	BP37
VSS	BP40
VSS	BP46
VSS	BP52
VSS	BP55
VSS	BP58
VSS	BP59
VSS	BP61
VSS	BP62
VSS	BP64
VSS	BP66
VSSA_USB	AU44
VSSA_USB	AW44
VSSCPUVIDSI0_1P03_SENSE	AF2
VSSRAMCPUSI1_1P03_SENSE	AF34

Table 35-2. Alphabetical Ball Listing

Ball	Signal
A5	VSS
A6	DDR3_1_MON2N
A8	DDR3_1_MON2P
A9	VSS
A12	VSS
A15	VSS
A18	VSS
A21	DDR3_1_DQS[2]
A24	VSS
A27	DDR3_1_DQ[10]
A30	VSS
A33	VSS
A36	DDR3_1_DQ[6]
A39	DDR3_1_DQ[4]
A42	GBE_TXN[3]
A45	VSS
A48	VSS
A51	VSS
A54	SATA_TXN[2]
A57	VSS
A59	VSS
A61	VSS
A62	VSS
A64	VSS
A66	VSS
B10	VSS
B12	DDR3_1_DQECC[7]
B14	DDR3_1_DQSECC[0]
B17	DDR3_1_DQECC[5]
B19	DDR3_1_DQ[19]
B21	DDR3_1_DQ[22]
B23	DDR3_1_DQ[17]
B26	VSS
B28	DDR3_1_DQ[15]
B30	DDR3_1_DQSB[1]
B32	DDR3_1_DQ[8]
B35	DDR3_1_DQ[2]
B37	DDR3_1_DQSB[0]
B39	DDR3_1_DQ[0]
B41	VSS
B44	GBE_TXN[2]
B46	GBE_TXN[1]
B48	GBE_TXN[0]
B50	GBE_REFCLKP
B53	SATA_TXN[3]
B55	VSS
B57	SATA_TXP[0]
C3	VSS
C5	DDR3_1_MON1N
C6	VSS
C9	DDR3_1_CK[0]
C13	DDR3_1_DQECC[6]
C15	DDR3_1_DQECC[1]
C18	VSS
C22	DDR3_1_DQSB[2]
C24	DDR3_1_DQ[20]
C27	DDR3_1_DQ[11]
C31	DDR3_1_DQ[9]
C33	VSS
C36	DDR3_1_DQ[7]
C40	DDR3_1_DQ[5]
C42	GBE_TXP[3]
C45	VSS
C49	VSS
C51	VSS
C54	SATA_TXP[2]
C58	VSS
C60	SATA_RXN[3]
C62	SATA_RXN[2]
C64	VSS
C66	VSS
D6	VSS
D8	DDR3_1_CKB[0]
D10	VSS
D12	DDR3_1_DQECC[2]
D14	DDR3_1_DQSBECC[0]
D15	DDR3_1_DQECC[0]
D17	DDR3_1_DQECC[4]
D19	DDR3_1_DQ[18]
D21	DDR3_1_DQ[23]
D23	DDR3_1_DQ[16]
D24	DDR3_1_DQ[21]
D26	VSS
D28	DDR3_1_DQ[14]
D30	DDR3_1_DQS[1]
D32	DDR3_1_DQ[13]
D33	VSS
D35	DDR3_1_DQ[3]
D37	DDR3_1_DQS[0]
D39	DDR3_1_DQ[1]
D41	VSS
D42	VSS
D44	GBE_TXP[2]
D46	GBE_TXP[1]
D48	GBE_TXP[0]
D50	GBE_REFCLKN
D51	VSS
D53	SATA_TXP[3]
D55	VSS
D57	SATA_TXN[0]
D59	VSS
D60	SATA_RXP[3]
E1	VSS
E3	VSS
E5	DDR3_1_MON1P
E7	VCCDDR_1_1P5
E9	VCCDDR_1_1P5
E10	VSS
E12	DDR3_1_DQECC[3]
E13	VSS
E14	VSS
E16	VSS
E18	VSS
E19	VSS
E21	VSS
E22	VSS
E23	VSS
E25	VSS
E27	VSS
E28	VSS
E30	VSS
E31	VSS
E32	DDR3_1_DQ[12]

Ball	Signal
E34	VSS
E36	VSS
E37	VSS
E39	VSS
E40	VSS
E41	VSS
E43	VSS
E45	VSS
E46	VSS
E48	VSS
E49	VSS
E50	VSS
E52	VSS
E54	VSS
E55	VSS
E57	SATA_TXN[1]
E58	SATA_TXP[1]
E59	VSS
E61	VSS
E62	SATA_RXP[2]
E64	VSS
E66	VSS
F1	VSS
F4	VSS
F5	VSS
F63	VSS
F66	VSS
G3	DDR3_1_MA[13]
G5	VCCDDR_1_1P5
G7	VCCDDR_1_1P5
G9	DDR3_1_CK[1]
G10	DDR3_1_CKB[2]
G17	DDR3_1_MA[1]
G18	DDR3_1_MA[2]
G25	DDR3_1_MA[12]
G26	DDR3_1_MA[14]
G32	VSS
G34	VSS
G41	VSS
G42	GBE_RXP[3]
G48	VSS

Ball	Signal
G50	GBE_OBSP
G56	VSS
G59	VSS
G62	SATA_RXP[1]
G64	SATA_RXN[1]
H1	VSS
H4	DDR3_1_CASB
H9	DDR3_1_CKB[1]
H10	DDR3_1_CK[2]
H13	VCCDDR_1_1P5
H14	VCCDDR_1_1P5
H17	DDR3_1_MA[0]
H18	DDR3_1_BS[1]
H21	DDR3_1_MA[8]
H22	DDR3_1_MA[5]
H25	DDR3_1_BS[2]
H26	DDR3_1_MA[15]
H29	DDR3_1_DQ[25]
H30	DDR3_1_DQ[24]
H32	DDR3_1_DQ[28]
H34	VSS
H37	VSS
H38	VSS
H41	VSS
H42	GBE_RXN[3]
H44	VSS
H46	VSS
H48	GBE_RXP[0]
H50	GBE_OBSN
H52	VSS
H54	VSS
H56	VSS
H58	VSS
H59	VSS
H62	SATA_RXP[0]
H63	SATA_RXN[0]
H66	VSS
J5	VCCDDR_1_1P5
J13	DDR3_1_CK[3]
J14	DDR3_1_REFP
J21	DDR3_1_MA[6]

Ball	Signal
J22	DDR3_1_MA[7]
J29	VSS
J30	DDR3_1_DQ[29]
J37	HPLL_REFN
J38	J38_RSVD
J44	GBE_RXP[2]
J46	GBE_RXP[1]
J52	SATA_OBSP
J54	SATA_REFCLKP
J63	VSS
J64	VSS
J66	VSS
K1	VSS
K3	DDR3_1_CSB[0]
K4	DDR3_1_CSB[3]
K10	VCCDDR_1_1P5
K17	VCCDDR_1_1P5
K18	VCCDDR_1_1P5
K25	VCCDDR_1_1P5
K26	VCCDDR_1_1P5
K32	VSS
K34	VSS
K41	VCCA_SATA_1P0
K42	VSS
K48	GBE_RXN[0]
K50	VSS
K62	VSS
L2	DDR3_1_ODT[0]
L4	DDR3_1_ODT[3]
L5	VCCDDR_1_1P5
L7	DDR3_1_CSB[2]
L8	DDR3_1_RASB
L13	DDR3_1_CKB[3]
L14	DDR3_1_REFN
L17	DDR3_1_DRAM_PWROK
L18	DDR3_1_MA[10]
L21	VCCDDR_1_1P5
L22	VCCDDR_1_1P5
L25	DDR3_1_CKE[2]
L26	DDR3_1_CKE[3]
L29	DDR3_1_DQSB[3]

Ball	Signal
L30	VSS
L32	DDR3_1_DQ[27]
L34	VSS
L37	HPLL_REFP
L38	L38_RSVD
L41	VCCA_SATA_1P0
L42	VCCA_SATA3_1P0
L44	GBE_RXN[2]
L46	GBE_RXN[1]
L48	VSS
L50	VSS
L52	SATA_OBSN
L54	SATA_REFCLKN
L57	SATA3_REFCLKP
L59	SATA3_REFCLKN
L60	VSS
L63	SATA3_RXP[1]
L65	SATA3_RXN[1]
M3	VCCDDR_1_1P5
M5	VSS
M21	DDR3_1_MA[4]
M22	DDR3_1_MA[9]
M29	DDR3_1_DQS[3]
M30	DDR3_1_DQ[30]
M37	VSS
M38	VSS
M44	VSS
M46	VSS
M52	VSS
M54	VSS
M56	VSS
M57	VSS
M59	VSS
M60	VSS
M62	SATA3_RXP[0]
M64	SATA3_RXN[0]
M66	VSS
N1	VSS
N2	DDR3_1_ODT[1]
N4	DDR3_1_CSB[1]
N5	VSS

Ball	Signal
N7	DDR3_1_WEB
N8	DDR3_1_ODT[2]
N10	VSS
N11	DDR3_1_DRAMRSTB
N13	VCCDDR_1_1P5
N14	VCCDDR_1_1P5
N17	DDR3_1_VCCA_PWROK
N18	DDR3_1_BS[0]
N25	DDR3_1_CKE[1]
N26	DDR3_1_CKE[0]
N32	DDR3_1_DQ[26]
N34	VSS
N41	VCCA_SATA_1P0
N42	VCCA_SATA3_1P0
N48	VSS
N50	VSS
N62	VSS
N63	VSS
N65	VSS
P17	VCCDDR_1_1P5
P18	VCCDDR_1_1P5
P21	DDR3_1_MA[3]
P22	DDR3_1_MA[11]
P25	VCCDDR_1_1P5
P26	VCCDDR_1_1P5
P29	VSS
P30	DDR3_1_DQ[31]
P32	VSS
P34	VSS
P37	VSS
P38	P38_RSVD
P41	VCCA_SATA_1P0
P42	VCCA_SATA3_1P0
P44	VSS
P46	GBE_LED0
P48	GBE_LED2
P50	GBE_SMBCLK
R2	VSS
R4	VSS
R5	VSS
R8	DDR3_1_DQ[38]

Ball	Signal
R9	DDR3_1_DQ[34]
R11	DDR3_1_DQ[35]
R12	VSS
R14	DDR3_1_DQ[33]
R15	DDR3_1_DQ[32]
R21	VCCDDR_1_1P5
R22	VCCDDR_1_1P5
R29	DDR3_1_VREF
R30	VSS
R37	R37_RSVD
R38	VSS
R44	VCCAPLL_SATA3_1P0
R46	VCCAPLL_SATA_1P0
R52	VSS
R53	SATA3_OBSN
R55	SATA3_OBSP
R56	VSS
R58	GBE_LED3
R59	GBE_EE_CS_N
R62	VSS
R63	SATA3_TXP[1]
R65	SATA3_TXN[1]
R66	SATA3_TXP[0]
T1	DDR3_1_DQ[45]
T3	DDR3_1_DQ[44]
T5	VSS
T17	DDR3_1_DQ[36]
T18	VSS
T20	VSS
T21	VSS
T23	VSS
T25	DDR3_1_CMDPU
T26	DDR3_1_DQPU
T28	VSS
T29	VSS
T31	VSS
T32	DDR3_1_ODTPU
T34	VSS
T36	VSS
T37	VCCA_GBE_1P0
T39	VCCAREF_GBE_HVGEN

Ball	Signal
T41	VSS
T42	VSS
T44	VCCAPLL_SATA3_1P0
T45	VCCAPLL_SATA_1P0
T47	VSS
T48	GBE_SDP0_1
T50	GBE_EE_SK
T52	VSS
T53	GPIO_SUS2
T55	GBE_SMBALRT_N
T56	VSS
T58	GBE_SDP0_0
T59	GBE_SMBD
T62	VSS
T64	SATA3_TXN[0]
U2	DDR3_1_DQ[41]
U4	DDR3_1_DQ[40]
U8	DDR3_1_DQ[39]
U9	VSS
U11	DDR3_1_DQS[4]
U12	DDR3_1_DQSB[4]
U14	VSS
U15	DDR3_1_DQ[37]
U62	VSS
U63	VSS
U65	VSS
V5	VSS
V20	VCCADLLDDR_1_1P0
V21	VCCADLLDDR_1_1P0
V23	VCCADLLDDR_1_1P0
V25	VCCADLLDDR_1_1P0
V26	VCCADLLDDR_1_1P0
V28	VCCACKDDR_1_1P0
V29	VCCCLKDDR_1_1P5
V31	VCCSFRPLLDDR_1_1P5
V32	VSS
V34	VCCSFRXXXSI0_1P35
V36	VSS
V38	VCCA_GBE_1P0
V39	VCCAREF_GBE_HVGEN
V41	VCCAPLL_GBE_1P0

Ball	Signal
V42	VSS
V44	VSS
V46	VCCAREF_SATA_HVGEN
V63	NCSI_RXD1
V64	PMU_SLP_S45_B
V66	GPIO_SUS0
W1	DDR3_1_DQS[5]
W3	DDR3_1_DQSB[5]
W4	VSS
W7	DDR3_1_DQSB[6]
W8	DDR3_1_DQ[54]
W10	DDR3_1_DQ[50]
W11	VSS
W13	DDR3_1_DQ[49]
W14	DDR3_1_DQ[48]
W16	VSS
W17	VSS
W19	VSS
W21	VCCADDR_1_1P0
W23	VCCADDR_1_1P0
W25	VCCADDR_1_1P0
W26	VCCADDR_1_1P0
W28	VCCACKDDR_1_1P0
W29	VCCCLKDDR_1_1P5
W31	VCCPLLDDR_1_1P0
W32	VSS
W34	VCCSFRXXXSI0_1P35
W36	VSS
W38	VCCA_GBE_1P0
W39	VSS
W41	VCCAPLL_GBE_1P0
W42	VSS
W44	VCCAREF_SATA_HVGEN
W46	VCCAREF_SATA_HVGEN
W48	VSS
W50	GBE_LED1
W51	GBE_EE_DI
W53	VSS
W54	GPIO_SUS1
W56	GBE_MDIO0_I2C_CLK
W57	VSS

Ball	Signal
W59	GBE_MDIO0_I2C_DATA
W60	GBE_EE_DO
W62	VSS
Y2	DDR3_1_DQ[47]
Y4	DDR3_1_DQ[46]
Y5	VSS
Y7	DDR3_1_DQS[6]
Y8	VSS
Y10	DDR3_1_DQ[55]
Y11	DDR3_1_DQ[51]
Y13	VSS
Y14	DDR3_1_DQ[53]
Y16	DDR3_1_DQ[52]
Y17	VSS
Y19	VSS
Y48	VSS
Y50	PMU_SLP_LAN_B
Y51	VSS
Y53	GBE_MDIO1_I2C_DATA
Y54	GBE_MDIO1_I2C_CLK
Y56	VSS
Y57	SUSPWRDNACK
Y59	NCSI_ARB_OUT
Y60	VSS
Y63	CPU_RESET_B
Y65	SPI_CS0_B
AA3	VSS
AA5	VSS
AA21	VCCADDR_1_1P0
AA23	VSS
AA25	VSS
AA26	VSS
AA28	VSS
AA29	VSS
AA31	VSS
AA32	VCCSFRXXXSI0_1P35
AA34	VCCSFRXXXSI0_1P35
AA36	VSS
AA38	VCCDIGXXXSUS_1P03
AA39	VCCDIGXXXSUS_1P03
AA41	VCCDIGXXXSUS_1P03

Ball	Signal
AA42	VSS
AA44	VSS
AA46	TRST_B
AA47	AA47_RSVD
AA62	VSS
AA64	VSS
AA66	VSS
AB1	VSS
AB2	DDR3_1_DQ[42]
AB4	DDR3_1_DQ[43]
AB5	VSS
AB20	VSS
AB62	SPI_MOSI
AB63	AB63_RSVD
AB65	SUS_STAT_B
AC8	DDR3_1_DQ[56]
AC9	DDR3_1_DQ[61]
AC11	DDR3_1_DQ[60]
AC12	VSS
AC14	DDR3_1_DQ[62]
AC15	DDR3_1_DQ[63]
AC17	DDR3_1_DQ[59]
AC18	VSS
AC20	VSS
AC21	VSS
AC23	VSS
AC25	AC25_RSVD
AC26	AC26_RSVD
AC28	VSS
AC29	VSS
AC31	VSS
AC32	VSS
AC34	VSS
AC36	VSS
AC38	VCCDIGXXXSUS_1P03
AC39	VCCDIGXXXSUS_1P03
AC41	VCCDIGXXXSUS_1P03
AC42	VCCFHVSOCSI0_1P03
AC44	VCCFHVSOCSI0_1P03
AC46	VSS
AC47	RSMRST_B

Ball	Signal
AC49	PMU_PWRBTN_B
AC50	VSS
AC52	PMU_SLP_DDRVTT_B
AC53	TMS
AC55	VSS
AC56	TDI
AC58	SPI_CS1_B
AC59	VSS
AD2	VSS
AD4	VSS
AD5	VSS
AD8	DDR3_1_DQ[57]
AD9	VSS
AD11	DDR3_1_DQSB[7]
AD12	DDR3_1_DQS[7]
AD14	VSS
AD15	DDR3_1_DQ[58]
AD17	VSS
AD18	VSS
AD20	VSS
AD21	VCCCPUVIDSI0_1P03
AD23	VCCCPUVIDSI0_1P03
AD25	VCCCPUVIDSI0_1P03
AD26	VCCCPUVIDSI0_1P03
AD28	VCCFHVCPUSI0_MOD0_1P03
AD29	VCCFHVCPUSI0_MOD2_1P03
AD31	VCCRAMCPUSI1_1P03
AD32	VCCRAMCPUSI1_1P03
AD34	VSS
AD36	VNN
AD38	VCCDIGXXXSI0_1P03
AD39	VNN
AD41	VSS
AD42	VCCPADXXXSUS_3P3
AD44	VCCPADXXXSUS_3P3
AD46	VSS
AD47	VSS
AD49	SRTCRST_B
AD50	RTEST_B
AD52	VSS

Ball	Signal
AD53	AD53_RSVD
AD55	BVCCRTC_EXTPAD
AD56	VSS
AD58	PMU_SUSCLK
AD59	SPI_MISO
AD62	VSS
AD63	USB_OC0_B
AD65	TCK
AD66	PMU_WAKE_B
AE1	VCCCPUVIDSI0_1P03
AE3	VCCCPUVIDSI0_1P03_SENSE
AE5	VCCCPUVIDSI0_1P03
AE47	VSS
AE62	PMU_PLTRST_B
AE64	VSS
AF2	VSSCPUVIDSI0_1P03_SENSE
AF4	VCCCPUVIDSI0_1P03
AF20	VCCCPUVIDSI0_1P03
AF21	VCCCPUVIDSI0_1P03
AF23	VCCCPUVIDSI0_1P03
AF25	VCCCPUVIDSI0_1P03
AF26	VCCCPUVIDSI0_1P03
AF28	VSS
AF29	VCCRAMCPUSI1_1P03
AF31	VCCRAMCPUSI1_1P03
AF32	VCCRAMCPUSI1_1P03_SENSE
AF34	VSSRAMCPUSI1_1P03_SENSE
AF36	VNN
AF38	VCCDIGXXXSI0_1P03
AF39	VNN
AF41	VCCPADXXXSUS_1P8
AF42	VCCPADXXXSUS_1P8
AF44	VSS
AF46	SPI_CLK
AF62	VSS
AF63	TDO
AF65	PMU_SLP_S3_B
AG5	VSS
AG7	VCCCPUVIDSI0_1P03

Ball	Signal
AG8	VCCCPUVIDSI0_1P03
AG10	VSS
AG11	VCCCPUVIDSI0_1P03
AG13	VCCCPUVIDSI0_1P03
AG14	VCCCPUVIDSI0_1P03
AG16	VSS
AG17	VCCCPUVIDSI0_1P03
AG19	VCCCPUVIDSI0_1P03
AG21	VCCCPUVIDSI0_1P03
AG23	VSS
AG25	VSS
AG26	VSS
AG28	VSS
AG29	VCCRAMCPUSI1_1P03
AG31	VCCRAMCPUSI1_1P03
AG32	VCCRAMCPUSI1_1P03
AG34	VSS
AG36	VNN
AG38	VCCDIGXXXSI0_1P03
AG39	VNN
AG41	VSS
AG42	VCCRTC_3P3
AG44	VSS
AG46	RCOMP_CORE_LVT
AG48	VSS
AG50	UART1_RXD
AG51	LPC_CLKOUT0
AG53	VSS
AG54	LPC_AD0
AG56	FLEX_CLK_SE1
AG57	VSS
AG59	LPC_AD3
AG60	AG60_RSVD
AG63	VSS
AG64	VSS
AG66	VSS
AH1	VCCCPUVIDSI0_1P03
AH3	VCCCPUVIDSI0_1P03
AH4	VCCCPUVIDSI0_1P03
AH7	VCCCPUVIDSI0_1P03
AH8	VCCCPUVIDSI0_1P03

Ball	Signal
AH10	VCCCPUVIDSI0_1P03
AH11	VSS
AH13	VCCCPUVIDSI0_1P03
AH14	VCCCPUVIDSI0_1P03
AH16	VCCCPUVIDSI0_1P03
AH17	VSS
AH19	VCCCPUVIDSI0_1P03
AH48	LPC_CLKRUNB
AH50	UART1_TXD
AH51	SATA3_GP0
AH53	VSS
AH54	SATA3_LEDN
AH56	LPC_FRAMEB
AH57	VSS
AH59	FLEX_CLK_SE0
AH60	COREPWROK
AH62	VSS
AJ2	VCCCPUVIDSI0_1P03
AJ4	VCCCPUVIDSI0_1P03
AJ5	VCCCPUVIDSI0_1P03
AJ21	VCCCPUVIDSI0_1P03
AJ23	VCCCPUVIDSI0_1P03
AJ25	VCCCPUVIDSI0_1P03
AJ26	VCCCPUVIDSI0_1P03
AJ28	VSS
AJ29	VCCFHVCPUSI0_MOD3_1P03
AJ31	VCCRAMCPUSI0GT_MOD3_1P03
AJ32	VSS
AJ34	AJ34_RSVD
AJ36	VNN
AJ38	VCCDIGXXXSI0_1P03
AJ39	VNN
AJ41	VCCPADXXXSI0_1P8
AJ42	VCCPADXXXSI0_3P3
AJ44	VSS
AJ46	VSS
AJ47	VSS
AJ63	BRTCX2_PAD
AJ65	BRTCX1_PAD
AK3	VSS

Ball	Signal
AK5	VSS
AK20	VCCCPUVIDSI0_1P03
AK62	VSS
AK64	VSS
AK66	VSS
AL1	VSS
AL2	DDR3_0_DQ[28]
AL4	DDR3_0_DQ[29]
AL5	VSS
AL8	DDR3_0_DQ[2]
AL9	VSS
AL11	DDR3_0_DQS[0]
AL12	DDR3_0_DQSB[0]
AL14	VSS
AL15	DDR3_0_DQ[5]
AL17	VSS
AL18	VSS
AL20	VSS
AL21	VCCCPUVIDSI0_1P03
AL23	VCCCPUVIDSI0_1P03
AL25	VCCCPUVIDSI0_1P03
AL26	VCCCPUVIDSI0_1P03
AL28	VCCFHVCPUSI0_MOD1_1P03
AL29	VSS
AL31	VCCCORE7VIDSI0GT_1P03
AL32	VCCCORE6VIDSI0GT_1P03
AL34	AL34_RSVD
AL36	VNN
AL38	VSS
AL39	VNN
AL41	VCCPADXXXSI0_1P8
AL42	VSS
AL44	VSS
AL46	MEMHOT_B
AL47	CTBTRIGOUT
AL49	SATA_LEDN
AL50	VSS
AL52	MCERR_B
AL53	LPC_AD2
AL55	VSS

Ball	Signal
AL56	NMI
AL58	SMBALRT_N0
AL59	VSS
AL62	ERROR1_B
AL63	ERROR2_B
AL65	ERROR0_B
AM8	DDR3_0_DQ[3]
AM9	DDR3_0_DQ[7]
AM11	DDR3_0_DQ[6]
AM12	VSS
AM14	DDR3_0_DQ[1]
AM15	DDR3_0_DQ[0]
AM17	DDR3_0_DQ[4]
AM18	VSS
AM20	VSS
AM21	VSS
AM23	VSS
AM25	VNN
AM26	VNN
AM28	VNN
AM29	VNN
AM31	VNN
AM32	VNN
AM34	VNN
AM36	VNN
AM38	VNN
AM39	VNN
AM41	VNN
AM42	VNN
AM44	VSS
AM46	VSS
AM47	CTBTRIGINOUT
AM49	LPC_CLKOUT1
AM50	VSS
AM52	IERR_B
AM53	LPC_AD1
AM55	VSS
AM56	CLK14_IN
AM58	PMU_RESETBUTTON_B
AM59	VSS
AN2	DDR3_0_DQ[25]

Ball	Signal
AN4	DDR3_0_DQ[24]
AN5	VSS
AN47	VSS
AN62	SMB_CLK0
AN63	SMB_DATA1
AN65	SMB_DATA2
AN66	VSS
AP1	DDR3_0_DQSB[3]
AP3	DDR3_0_DQS[3]
AP5	VSS
AP20	AP20_RSVD
AP21	AP21_RSVD
AP23	VSS
AP25	VSS
AP26	VSS
AP28	VSS
AP29	VSS
AP31	VSS
AP32	VSS
AP34	VSS
AP36	VSS
AP38	VSS
AP39	VSS
AP41	VSS
AP42	VSS
AP44	VSS
AP46	VSS
AP62	SMB_DATA0
AP64	VSS
AR2	DDR3_0_DQ[30]
AR4	DDR3_0_DQ[31]
AR7	DDR3_0_DQSB[1]
AR8	VSS
AR10	DDR3_0_DQ[15]
AR11	DDR3_0_DQ[11]
AR13	VSS
AR14	DDR3_0_DQ[8]
AR16	DDR3_0_DQ[12]
AR17	VSS
AR19	VSS
AR48	USB_OBSP

Ball	Signal
AR50	VSS
AR51	AR51_RSVD
AR53	AR53_RSVD
AR54	AR54_RSVD
AR56	VSS
AR57	DFX_PORT3
AR59	DFX_PORT15
AR60	VSS
AR62	VSS
AR63	SMB_CLK1
AR65	SMB_CLK2
AT5	VSS
AT7	DDR3_0_DQS[1]
AT8	DDR3_0_DQ[14]
AT10	DDR3_0_DQ[10]
AT11	VSS
AT13	DDR3_0_DQ[9]
AT14	DDR3_0_DQ[13]
AT16	VSS
AT17	VSS
AT19	VSS
AT21	VCCADDR_0_1P0
AT23	VCCADDR_0_1P0
AT25	VCCADDR_0_1P0
AT26	VCCADDR_0_1P0
AT28	VCCADDR_0_1P0
AT29	VCCCLKDDR_0_1P5
AT31	VCCSFRPLLDDR_0_1P5
AT32	VCCACKDDR_0_1P0
AT34	AT34_RSVD
AT36	VCCA_PCIE_1P0
AT38	VCCA_PCIE_1P0
AT39	VCCAREF_PCIE_HVGEN
AT41	VCCDUSBSUS_1P0
AT42	VCCAUSB_1P0
AT44	VSS
AT46	USB_RCOMPI
AT48	USB_RCOMPO
AT50	ILB_SERIRQ
AT51	AT51_RSVD
AT53	VSS

Ball	Signal
AT54	DFX_PORT4
AT56	THERMTRIP_N
AT57	VSS
AT59	DFX_PORT11
AT60	DFX_PORT9
AT63	SATA_GP0
AT64	VSS
AT66	VSS
AU1	DDR3_0_DQ[26]
AU3	DDR3_0_DQ[27]
AU4	VSS
AU21	VCCADLLDDR_0_1P0
AU23	VCCADLLDDR_0_1P0
AU25	VCCADLLDDR_0_1P0
AU26	VCCADLLDDR_0_1P0
AU28	VCCADLLDDR_0_1P0
AU29	VCCCLKDDR_0_1P5
AU31	VCCPLLDDR_0_1P0
AU32	VCCACKDDR_0_1P0
AU34	AU34_RSVD
AU36	VCCA_PCIE_1P0
AU38	VCCA_PCIE_1P0
AU39	VCCA_PCIE_1P0
AU41	VCCDUSB_1P0
AU42	VCCUSBSUS_3P3
AU44	VSSA_USB
AU46	VCCUSBSUS_1P8
AU47	VCCUSBSUS_1P8
AU62	VSS
AV2	VSS
AV4	VSS
AV5	VSS
AV63	DFX_PORT_CLK0
AV65	DFX_PORT_CLK1
AW3	DDR3_0_DQECC[4]
AW5	VSS
AW8	DDR3_0_DQ[20]
AW9	VSS
AW11	DDR3_0_DQSB[2]
AW12	DDR3_0_DQS[2]
AW14	VSS

Ball	Signal
AW15	DDR3_0_DQ[23]
AW17	DDR3_0_DQ[19]
AW19	VSS
AW20	DDR3_0_ODTPU
AW21	DDR3_0_DQPU
AW23	VSS
AW25	VSS
AW26	VSS
AW28	VSS
AW30	VSS
AW31	VSS
AW32	VSS
AW34	VSS
AW36	VCCA_PCIE_1P0
AW38	VCCA_PCIE_1P0
AW39	VCCA_PCIE_1P0
AW41	VCCDUSB_1P0
AW42	VCCUSBSUS_3P3
AW44	VSSA_USB
AW46	VSS
AW47	VSS
AW49	VSS
AW50	VSS
AW52	VSS
AW53	SVID_DATA
AW55	VSS
AW56	CX_PRDY_B
AW58	DFX_PORT2
AW59	VSS
AW62	DFX_PORT7
AW64	DFX_PORT12
AW66	DFX_PORT13
AY1	DDR3_0_DQECC[5]
AY2	DDR3_0_DQECC[1]
AY4	DDR3_0_DQECC[0]
AY5	VSS
AY8	DDR3_0_DQ[21]
AY9	DDR3_0_DQ[16]
AY11	DDR3_0_DQ[17]
AY12	VSS
AY14	DDR3_0_DQ[22]

Ball	Signal
AY15	DDR3_0_DQ[18]
AY19	VCCDDR_0_1P5
AY21	VCCDDR_0_1P5
AY23	VCCDDR_0_1P5
AY29	DDR3_0_VREF
AY30	DDR3_0_DQ[37]
AY37	VCCAPLL_PCIE_1P0
AY38	VSS
AY45	VSS
AY46	USB_DP[0]
AY52	VSS
AY53	CX_PREQ_B
AY55	VSS
AY56	DFX_PORT5
AY58	DFX_PORT1
AY59	DFX_PORT8
AY62	VSS
AY63	DFX_PORT6
AY65	DFX_PORT14
BA17	VSS
BA21	DDR3_0_CK[0]
BA23	DDR3_0_MA[13]
BA25	DDR3_0_DRAMRSTB
BA26	DDR3_0_CMDPU
BA29	VSS
BA30	DDR3_0_DQ[33]
BA32	DDR3_0_DQ[32]
BA34	VSS
BA37	VCCAPLL_PCIE_1P0
BA38	PCIE_OBSN
BA41	USB_DN[3]
BA42	VSS
BA45	USB_DP[1]
BA46	USB_DN[0]
BA49	VSS
BA50	USB_REFCLKN
BB2	DDR3_0_DQSECC[0]
BB4	DDR3_0_DQSBECC[0]
BB5	VSS
BB17	VSS
BB19	DDR3_0_CK[3]

Ball	Signal
BB25	DDR3_0_CSB[1]
BB26	DDR3_0_ODT[1]
BB32	DDR3_0_DQ[36]
BB34	VSS
BB41	USB_DP[3]
BB42	USB_DN[2]
BB49	PCIE_REFCLKP
BB50	USB_REFCLKP
BB53	VSS
BB54	VSS
BB56	VSS
BB57	VSS
BB59	PROCHOT_B
BB60	SVID_CLK
BB62	VSS
BB63	DFX_PORT10
BB65	VSS
BB66	VSS
BC1	DDR3_0_DQECC[6]
BC3	DDR3_0_DQECC[7]
BC5	VSS
BC7	VCCDDR_0_1P5
BC8	DDR3_0_BS[2]
BC10	VCCDDR_0_1P5
BC11	DDR3_0_MA[14]
BC13	VCCDDR_0_1P5
BC15	VCCDDR_0_1P5
BC21	DDR3_0_CKB[0]
BC23	DDR3_0_WEB
BC29	DDR3_0_DQSB[4]
BC30	VSS
BC37	VSS
BC38	PCIE_OBSP
BC45	USB_DN[1]
BC46	VSS
BC62	SVID_ALERT_B
BC64	DFX_PORT0
BD2	DDR3_0_DQECC[2]
BD4	DDR3_0_DQECC[3]
BD7	DDR3_0_MA[12]
BD8	DDR3_0_MA[8]

Ball	Signal
BD10	DDR3_0_MA[6]
BD13	DDR3_0_MA[4]
BD15	DDR3_0_MA[9]
BD17	VCCDDR_0_1P5
BD19	DDR3_0_CKB[3]
BD21	VCCDDR_0_1P5
BD23	DDR3_0_CSB[2]
BD25	VCCDDR_0_1P5
BD26	DDR3_0_ODT[0]
BD29	DDR3_0_DQS[4]
BD30	DDR3_0_DQ[35]
BD32	VSS
BD34	VSS
BD37	PCIE_RXP[14]
BD38	VSS
BD41	VSS
BD42	USB_DP[2]
BD45	VSS
BD46	PCIE_RXP[9]
BD49	PCIE_REFCLKN
BD50	VSS
BD53	VSS
BD54	PCIE_RXP[5]
BD59	VSS
BD60	VSS
BD62	VSS
BD63	VSS
BD65	VSS
BE5	VSS
BE17	DDR3_0_VCCA_PWROK
BE19	VCCDDR_0_1P5
BE25	DDR3_0_CSB[0]
BE26	VCCDDR_0_1P5
BE32	VSS
BE34	PCIE_RXP[15]
BE41	PCIE_RXP[12]
BE42	VSS
BE49	VSS
BE50	PCIE_RXP[7]
BE57	VSS
BE63	PCIE_RXP[0]

Ball	Signal
BE64	PCIE_RXN[0]
BE66	VSS
BF1	VSS
BF3	VSS
BF4	VSS
BF13	VCCDDR_0_1P5
BF15	DDR3_0_MA[15]
BF21	DDR3_0_CK[1]
BF23	VCCDDR_0_1P5
BF29	VSS
BF30	DDR3_0_DQ[34]
BF37	PCIE_RXN[14]
BF38	PCIE_RXP[13]
BF45	PCIE_RXP[10]
BF46	PCIE_RXN[9]
BF53	PCIE_RXP[6]
BF54	PCIE_RXN[5]
BF62	VSS
BG1	VSS
BG4	DDR3_0_CKE[0]
BG5	DDR3_0_CKE[2]
BG8	DDR3_0_MA[11]
BG9	VCCDDR_0_1P5
BG11	DDR3_0_MA[3]
BG13	DDR3_0_MA[2]
BG15	VCCDDR_0_1P5
BG17	DDR3_0_DRAM_PWROK
BG19	DDR3_0_CK[2]
BG21	DDR3_0_CKB[1]
BG23	DDR3_0_RASB
BG25	DDR3_0_ODT[3]
BG26	DDR3_0_CSB[3]
BG29	DDR3_0_DQ[38]
BG30	DDR3_0_DQ[39]
BG32	VSS
BG34	PCIE_RXN[15]
BG37	VSS
BG38	PCIE_RXN[13]
BG41	PCIE_RXN[12]
BG42	PCIE_RXP[11]
BG45	PCIE_RXN[10]

Ball	Signal
BG46	VSS
BG49	PCIE_RXP[8]
BG50	PCIE_RXN[7]
BG53	PCIE_RXN[6]
BG54	VSS
BG57	PCIE_RXP[4]
BG58	PCIE_RXP[3]
BG63	VSS
BG66	VSS
BH3	DDR3_0_CKE[3]
BH5	DDR3_0_CKE[1]
BH8	DDR3_0_MA[7]
BH11	DDR3_0_MA[0]
BH17	VCCDDR_0_1P5
BH19	DDR3_0_CKB[2]
BH25	DDR3_0_ODT[2]
BH26	DDR3_0_CASB
BH32	VSS
BH34	VSS
BH41	VSS
BH42	PCIE_RXN[11]
BH49	PCIE_RXN[8]
BH50	VSS
BH57	PCIE_RXN[4]
BH58	PCIE_RXN[3]
BH60	VSS
BH62	VSS
BH64	PCIE_RXN[1]
BJ1	VSS
BJ4	VCCDDR_0_1P5
BJ62	PCIE_RXP[2]
BJ63	PCIE_RXP[1]
BJ66	VSS
BK1	VSS
BK3	DDR3_0_MON2N
BK5	DDR3_0_MON2P
BK6	VCCDDR_0_1P5
BK8	VSS
BK9	VCCDDR_0_1P5
BK10	VSS
BK12	VCCDDR_0_1P5

Ball	Signal
BK13	VCCDDR_0_1P5
BK15	VSS
BK17	VSS
BK18	VSS
BK19	VSS
BK21	VSS
BK22	VSS
BK24	VSS
BK26	VSS
BK27	VSS
BK28	VSS
BK30	VSS
BK31	VSS
BK32	DDR3_0_DQ[57]
BK35	DDR3_0_DQ[62]
BK36	VSS
BK37	VSS
BK39	VSS
BK40	PCIE_TXN[14]
BK42	VSS
BK44	VSS
BK45	VSS
BK46	VSS
BK48	VSS
BK49	VSS
BK51	VSS
BK53	VSS
BK54	PCIE_TXN[4]
BK56	VSS
BK57	VSS
BK58	VSS
BK60	PCIE_TXP[0]
BK62	PCIE_RXN[2]
BK64	VSS
BK66	VSS
BL7	VCCDDR_0_1P5
BL8	DDR3_0_MA[1]
BL10	DDR3_0_BS[1]
BL12	DDR3_0_BS[0]
BL14	VSS
BL16	DDR3_0_DQ[40]

Ball	Signal
BL17	DDR3_0_DQ[41]
BL19	VSS
BL21	DDR3_0_DQ[42]
BL23	DDR3_0_DQ[52]
BL25	VSS
BL26	DDR3_0_DQSB[6]
BL28	DDR3_0_DQ[50]
BL30	VSS
BL32	DDR3_0_DQ[61]
BL34	VSS
BL35	DDR3_0_DQ[63]
BL37	VSS
BL39	PCIE_TXN[15]
BL41	PCIE_TXP[13]
BL43	VSS
BL44	PCIE_TXN[11]
BL46	PCIE_TXN[10]
BL48	PCIE_TXN[9]
BL50	PCIE_TXN[7]
BL52	PCIE_TXP[6]
BL53	PCIE_TXN[5]
BL55	PCIE_TXP[3]
BL57	PCIE_TXP[2]
BL59	PCIE_TXP[1]
BL61	PCIE_TXN[0]
BM1	VSS
BM3	VSS
BM5	VSS
BM7	DDR3_0_MA[5]
BM9	VCCDDR_0_1P5
BM13	DDR3_0_REFN
BM16	DDR3_0_DQ[44]
BM18	DDR3_0_DQS[5]
BM22	VSS
BM25	DDR3_0_DQ[48]
BM27	DDR3_0_DQ[54]
BM31	DDR3_0_DQ[60]
BM34	DDR3_0_DQSB[7]
BM36	DDR3_0_DQ[59]
BM40	PCIE_TXP[14]
BM43	PCIE_TXN[12]

Ball	Signal
BM45	VSS
BM49	PCIE_TXN[8]
BM52	PCIE_TXN[6]
BM54	PCIE_TXP[4]
BM58	PCIE_TXN[1]
BM61	VSS
BM62	VSS
BM64	VSS
BM66	VSS
BN10	DDR3_0_MA[10]
BN12	DDR3_0_REFP
BN14	VSS
BN17	DDR3_0_DQSB[5]
BN19	DDR3_0_DQ[47]
BN21	DDR3_0_DQ[43]
BN23	DDR3_0_DQ[53]
BN26	DDR3_0_DQS[6]
BN28	DDR3_0_DQ[51]
BN30	VSS
BN32	DDR3_0_DQ[56]
BN35	DDR3_0_DQ[58]
BN37	VSS
BN39	PCIE_TXP[15]
BN41	PCIE_TXN[13]
BN44	PCIE_TXP[11]
BN46	PCIE_TXP[10]
BN48	PCIE_TXP[9]
BN50	PCIE_TXP[7]
BN53	PCIE_TXP[5]
BN55	PCIE_TXN[3]
BN57	PCIE_TXN[2]
BP1	VSS
BP3	VSS
BP5	VSS
BP6	DDR3_0_MON1N
BP8	DDR3_0_MON1P
BP10	VSS
BP13	VSS
BP16	DDR3_0_DQ[45]
BP19	DDR3_0_DQ[46]
BP22	VSS

Ball	Signal
BP25	DDR3_0_DQ[49]
BP28	DDR3_0_DQ[55]
BP31	VSS
BP34	DDR3_0_DQS[7]
BP37	VSS
BP40	VSS
BP43	PCIE_TXP[12]
BP46	VSS
BP49	PCIE_TXP[8]
BP52	VSS
BP55	VSS
BP58	VSS
BP59	VSS
BP61	VSS
BP62	VSS
BP64	VSS
BP66	VSS

35.1 Ball Map

The ball map is divided into eight sections. The sections are defined in Table 35-3. The eight sections are shown as eight separate tables, from Table 35-4 through Table 35-7.

Table 35-3. Ball Map

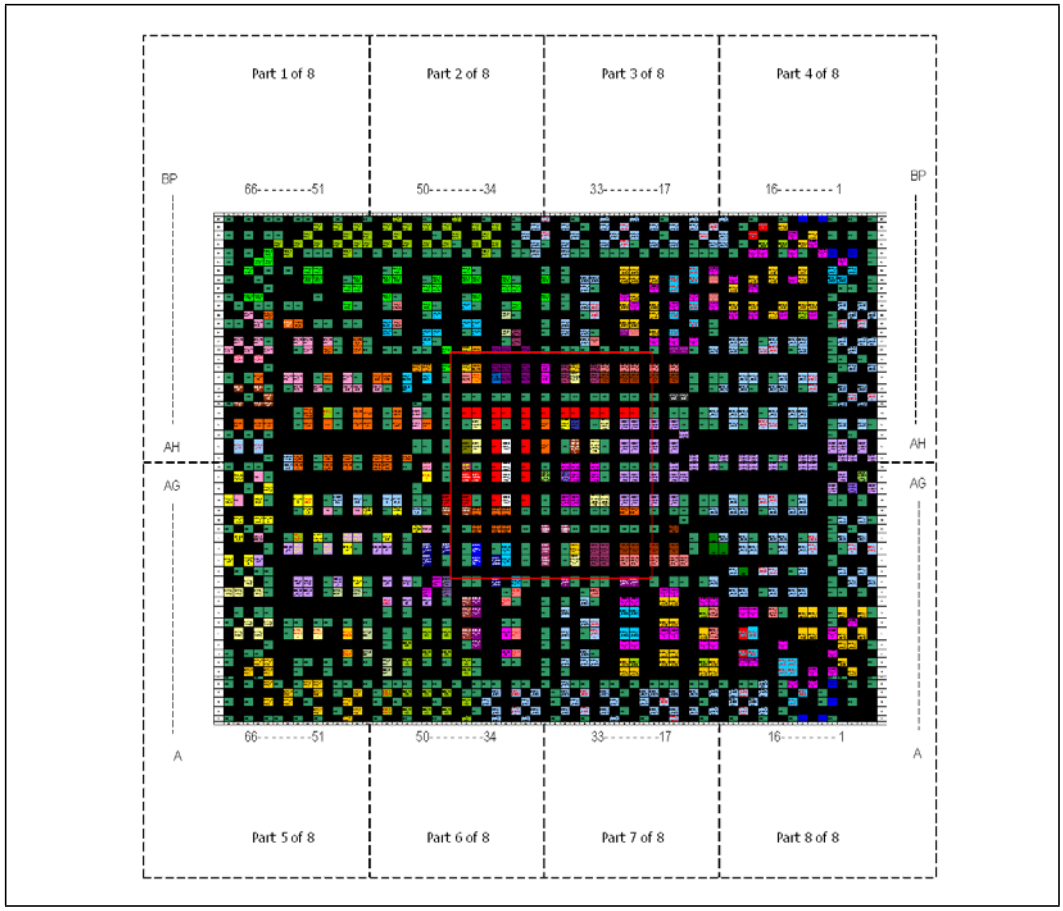


Table 35-4. Top Left (Sheet 1 of 5)

	BP	BN	BM	BL	BK	BJ	BH	BG	BF	BE	BD	BC	BB	BA	AY	AW	AV	AU	AT	AR	AP	AN	AM	AL	AK	AJ	AH
66	VSS		VSS		VSS	VSS		VSS		VSS			VSS			DFX_PORT13			VSS			VSS			VSS		
65											VSS		VSS		DFX_PORT14		DFX_PORT_CLK1			SMB_CLK2		SMB_DATA2		ERROR0_B		BRTCX1_PAD	
64	VSS		VSS		VSS		PCIE_RXN[1]			PCIE_RXN[0]		DFX_PORT0				DFX_PORT12			VSS		VSS				VSS		
63						PCIE_RXP[1]		VSS		PCIE_RXP[0]	VSS		DFX_PORT10		DFX_PORT6		DFX_PORT_CLK0		SATA_GP0	SMB_CLK1		SMB_DATA1		ERROR2_B		BRTCX2_PAD	
62	VSS		VSS		PCIE_RXN[2]	PCIE_RXP[2]	VSS		VSS		VSS	SVID_ALERT_B	VSS		VSS	DFX_PORT7		VSS		VSS	SMB_DATA0	SMB_CLK0		ERROR1_B	VSS		VSS
61	VSS		VSS	PCIE_TXN[0]																							
60					PCIE_TXP[0]		VSS				VSS		SVID_CLK						DFX_PORT9	VSS							COREPWROK
59	VSS			PCIE_TXP[1]							VSS		PROCHOT_B		DFX_PORT8	VSS			DFX_PORT11	DFX_PORT15			VSS	VSS			FLEX_CLK_SE0
58	VSS		PCIE_TXN[1]		VSS		PCIE_RXN[3]	PCIE_RXP[3]							DFX_PORT1	DFX_PORT2							PMU_RESETBUTTON_B	SMBALRT_N0			
57		PCIE_TXN[2]		PCIE_TXP[2]	VSS		PCIE_RXN[4]	PCIE_RXP[4]		VSS			VSS						VSS	DFX_PORT3							VSS
56					VSS								VSS		DFX_PORT5	CX_PRDY_B			THERMTRIP_N	VSS			CLK14_IN	NMI			LPC_FRAMEB
55	VSS	PCIE_TXN[3]		PCIE_TXP[3]											VSS	VSS							VSS	VSS			
54			PCIE_TXP[4]		PCIE_TXN[4]			VSS	PCIE_RXN[5]		PCIE_RXP[5]		VSS						DFX_PORT4	AR54_RSVD							SATA3_LEDN
53		PCIE_TXP[5]		PCIE_TXN[5]	VSS			PCIE_RXN[6]	PCIE_RXP[6]		VSS		VSS		CX_PREQ_B	SVID_DATA			VSS	AR53_RSVD			LPC_AD1	LPC_AD2			VSS
52	VSS		PCIE_TXN[6]	PCIE_TXP[6]											VSS	VSS							IERR_B	MCERR_B			
51					VSS														AT51_RSVD	AR51_RSVD							SATA3_GP0

Table 35-4. Top Left (Sheet 2 of 5)

	BP	BN	BM	BL	BK	BJ	BH	BG	BF	BE	BD	BC	BB	BA	AY	AW	AV	AU	AT	AR	AP	AN	AM	AL	AK	AJ	AH
50		PCIE_TXP[7]		PCIE_TXN[7]			VSS	PCIE_RXN[7]		PCIE_RXP[7]	VSS		USB_RCOMPO	USB_RFCLKN		VSS			ILB_SERIRQ	VSS			VSS	VSS			UART1_TXD
49	PCIE_TXP[8]		PCIE_TXN[8]		VSS		PCIE_RXN[8]	PCIE_RXP[8]		VSS	PCIE_RFCLKN		PCIE_RFCLKP	VSS		VSS							LPC_CLKOUT1	SATA_LEDN			
48		PCIE_TXP[9]		PCIE_TXN[9]	VSS														USB_RCOMPO	USB_OBSP							LPC_CLKRUNB
47																VSS		VCCUSBSUS_1P8				VSS	CTBTRIGINOUT	CTBTRIGOUT		VSS	
46	VSS	PCIE_TXP[10]		PCIE_TXN[10]	VSS			VSS	PCIE_RXN[9]		PCIE_RXP[9]	VSS		USB_DN[0]	USB_DP[0]	VSS		VCCUSBSUS_1P8	USB_RCOMPI		VSS		VSS	MEMHOT_B		VSS	
45			VSS		VSS			PCIE_RXN[10]	PCIE_RXP[10]		VSS	USB_DN[1]		USB_DP[1]	VSS												
44		PCIE_TXP[11]		PCIE_TXN[11]	VSS											VSSA_USB		VSSA_USB	VSS		VSS		VSS	VSS		VSS	
43	PCIE_TXP[12]		PCIE_TXN[12]	VSS																							
42					VSS		PCIE_RXN[11]	PCIE_RXP[11]		VSS	USB_DP[2]		USB_DN[2]	VSS		VCCUSBSUS_3P3		VCCUSBSUS_3P3	VCCAUSB_1P0		VSS		VNN	VSS		VCCPADXXXSI0_3P3	
41		PCIE_TXN[13]		PCIE_TXP[13]			VSS	PCIE_RXN[12]		PCIE_RXP[12]	VSS		USB_DP[3]	USB_DN[3]		VCCDUSB_1P0		VCCDUSB_1P0	VCCDUSBSUS_1P0		VSS		VNN	VCCPADXXXSI0_1P8		VCCPADXXXSI0_1P8	
40	VSS		PCIE_TXP[14]		PCIE_TXN[14]																						
39		PCIE_TXP[15]		PCIE_TXN[15]	VSS											VCCA_PCIE_1P0		VCCA_PCIE_1P0	VCCAREF_PCIE_HVGEN		VSS		VNN	VNN		VNN	
38								PCIE_RXN[13]	PCIE_RXP[13]		VSS	PCIE_OBSP		PCIE_OBSN	VSS	VCCA_PCIE_1P0		VCCA_PCIE_1P0	VCCA_PCIE_1P0		VSS		VNN	VSS		VCCDIGXXXSI0_1P03	
37	VSS	VSS		VSS	VSS			VSS	PCIE_RXN[14]		PCIE_RXP[14]	VSS		VCCAPLL_PCIE_1P0	VCCAPLL_PCIE_1P0											

Table 35-4. Top Left (Sheet 3 of 5)

	BP	BN	BM	BL	BK	BJ	BH	BG	BF	BE	BD	BC	BB	BA	AY	AW	AV	AU	AT	AR	AP	AN	AM	AL	AK	AJ	AH
36			DDR3_0_DQ[59]		VSS											VCCA_PCIE_1P0		VCCA_PCIE_1P0	VCCA_PCIE_1P0		VSS		VNN	VNN		VNN	
35		DDR3_0_DQ[58]		DDR3_0_DQ[63]	DDR3_0_DQ[62]																						
34	DDR3_0_DQS[7]		DDR3_0_DQSB[7]	VSS			VSS	PCIE_RXN[15]		PCIE_RXP[15]	VSS		VSS	VSS		VSS		AU34_RSVD	AT34_RSVD		VSS		VNN	AL34_RSVD		AJ34_RSVD	
33																											
32		DDR3_0_DQ[56]		DDR3_0_DQ[61]	DDR3_0_DQ[57]		VSS	VSS		VSS	VSS		DDR3_0_DQ[36]	DDR3_0_DQ[32]		VSS		VCCACKDDR_0_1P0	VCCACKDDR_0_1P0		VSS		VNN	VCCCORE6VIDSI0GT_1P03		VSS	
31	VSS		DDR3_0_DQ[60]		VSS											VSS		VCCPLLDDR_0_1P0	VCCSFRPLLDDR_0_1P5		VSS		VNN	VCCCORE7VIDSI0GT_1P03		VCCRAMCPUSI0GT_MOD3_1P03	
30		VSS		VSS	VSS			DDR3_0_DQ[39]	DDR3_0_DQ[34]		DDR3_0_DQ[35]	VSS		DDR3_0_DQ[33]	DDR3_0_DQ[37]	VSS											
29								DDR3_0_DQ[38]	VSS		DDR3_0_DQS[4]	DDR3_0_DQSB[4]		VSS	DDR3_0_VREF			VCCCLKDDR_0_1P5	VCCCLKDDR_0_1P5		VSS		VNN	VSS		VCCFHVCPUSI0_MOD3_1P03	
28	DDR3_0_DQ[55]	DDR3_0_DQ[51]		DDR3_0_DQ[50]	VSS											VSS		VCCADLLDDR_0_1P0	VCCADDR_0_1P0		VSS		VNN	VCCFHVCPUSI0_MOD1_1P03		VSS	
27			DDR3_0_DQ[54]		VSS																						
26		DDR3_0_DQS[6]		DDR3_0_DQSB[6]	VSS		DDR3_0_CASB	DDR3_0_CSB[3]		VCCDDR_0_1P5	DDR3_0_ODT[0]		DDR3_0_ODT[1]	DDR3_0_CMDPU		VSS		VCCADLLDDR_0_1P0	VCCADDR_0_1P0		VSS		VNN	VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	
25	DDR3_0_DQ[49]		DDR3_0_DQ[48]	VSS			DDR3_0_ODT[2]	DDR3_0_ODT[3]		DDR3_0_CSB[0]	VCCDDR_0_1P5		DDR3_0_CSB[1]	DDR3_0_DRAMRSTB		VSS		VCCADLLDDR_0_1P0	VCCADDR_0_1P0		VSS		VNN	VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	
24					VSS																						

Table 35-4. Top Left (Sheet 4 of 5)

	BP	BN	BM	BL	BK	BJ	BH	BG	BF	BE	BD	BC	BB	BA	AY	AW	AV	AU	AT	AR	AP	AN	AM	AL	AK	AJ	AH
23		DDR3_0_DQ[53]		DDR3_0_DQ[52]				DDR3_0_RASB	VCCDDR_0_1P5		DDR3_0_CSB[2]	DDR3_0_WEB		DDR3_0_MA[13]	VCCDDR_0_1P5	VSS		VCCADLLDDR_0_1P0	VCCADDR_0_1P0		VSS		VSS	VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	
22	VSS		VSS		VSS																						
21		DDR3_0_DQ[43]		DDR3_0_DQ[42]	VSS			DDR3_0_CKB[1]	DDR3_0_CK[1]		VCCDDR_0_1P5	DDR3_0_CKB[0]		DDR3_0_CK[0]	VCCDDR_0_1P5	DDR3_0_DQPU		VCCADLLDDR_0_1P0	VCCADDR_0_1P0		AP21_RSVD		VSS	VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	
20																DDR3_0_ODTPU					AP20_RSVD		VSS	VSS	VCCCPUVIDSI0_1P03		
19	DDR3_0_DQ[46]	DDR3_0_DQ[47]		VSS	VSS		DDR3_0_CKB[2]	DDR3_0_CK[2]		VCCDDR_0_1P5	DDR3_0_CKB[3]		DDR3_0_CK[3]		VCCDDR_0_1P5	VSS			VSS	VSS							VCCCPUVIDSI0_1P03
18			DDR3_0_DQS[5]		VSS																		VSS	VSS			
17		DDR3_0_DQSB[5]		DDR3_0_DQ[41]	VSS		VCCDDR_0_1P5	DDR3_0_DRAM_PWROK		DDR3_0_VCCA_PWROK	VCCDDR_0_1P5		VSS	VSS		DDR3_0_DQ[19]			VSS	VSS			DDR3_0_DQ[4]	VSS			VSS
16	DDR3_0_DQ[45]		DDR3_0_DQ[44]	DDR3_0_DQ[40]															VSS	DDR3_0_DQ[12]							VCCCPUVIDSI0_1P03
15					VSS			VCCDDR_0_1P5	DDR3_0_MA[15]		DDR3_0_MA[9]	VCCDDR_0_1P5			DDR3_0_DQ[18]	DDR3_0_DQ[23]							DDR3_0_DQ[0]	DDR3_0_DQ[5]			
14		VSS		VSS											DDR3_0_DQ[22]	VSS			DDR3_0_DQ[13]	DDR3_0_DQ[8]			DDR3_0_DQ[1]	VSS			VCCCPUVIDSI0_1P03
13	VSS		DDR3_0_REFN		VCCDDR_0_1P5			DDR3_0_MA[2]	VCCDDR_0_1P5		DDR3_0_MA[4]	VCCDDR_0_1P5							DDR3_0_DQ[9]	VSS							VCCCPUVIDSI0_1P03
12		DDR3_0_REFP		DDR3_0_BS[0]	VCCDDR_0_1P5										VSS	DDR3_0_DQS[2]							VSS	DDR3_0_DQSB[0]			
11							DDR3_0_MA[0]	DDR3_0_MA[3]				DDR3_0_MA[14]			DDR3_0_DQ[17]	DDR3_0_DQSB[2]			VSS	DDR3_0_DQ[11]			DDR3_0_DQ[6]	DDR3_0_DQS[0]			VSS
10	VSS	DDR3_0_MA[10]		DDR3_0_BS[1]	VSS						DDR3_0_MA[6]	VCCDDR_0_1P5							DDR3_0_DQ[10]	DDR3_0_DQ[15]							VCCCPUVIDSI0_1P03

Table 35-4. Top Left (Sheet 5 of 5)

	BP	BN	BM	BL	BK	BJ	BH	BG	BF	BE	BD	BC	BB	BA	AY	AW	AV	AU	AT	AR	AP	AN	AM	AL	AK	AJ	AH
9			VCC DDR _0_ 1P5		VCC DDR _0_ 1P5			VCC DDR _0_ 1P5							DDR 3_0 _DQ [16]	VSS							DDR 3_0 _DQ [7]	VSS			
8	DDR 3_0 _MO N1P			DDR 3_0 _MA [1]	VSS		DDR 3_0 _MA [7]	DDR 3_0 _MA [11]			DDR 3_0 _MA [8]	DDR 3_0 _BS[2]			DDR 3_0 _DQ [21]	DDR 3_0 _DQ [20]			DDR 3_0 _DQ [14]	VSS			DDR 3_0 _DQ [3]	DDR 3_0 _DQ [2]			VCC CPU VID SI0_ 1P03
7			DDR 3_0 _MA [5]	VCC DDR _0_ 1P5							DDR 3_0 _MA [12]	VCC DDR _0_ 1P5							DDR 3_0 _DQ S[1]	DDR 3_0 _DQ SB[1]							VCC CPU VID SI0_ 1P03
6	DDR 3_0 _MO N1N				VCC DDR _0_ 1P5																						
5	VSS		VSS		DDR 3_0 _MO N2P		DDR 3_0 _CK E[1]	DDR 3_0 _CK E[2]		VSS		VSS	VSS		VSS	VSS	VSS		VSS		VSS	VSS		VSS	VSS	VCC CPU VID SI0_ 1P03	
4						VCC DDR _0_ 1P5		DDR 3_0 _CK E[0]	VSS		DDR 3_0 _DQ ECC[3]		DDR 3_0 _DQ SBE CC[0]		DDR 3_0 _DQ ECC[0]		VSS	VSS		DDR 3_0 _DQ [31]		DDR 3_0 _DQ [24]		DDR 3_0 _DQ [29]		VCC CPU VID SI0_ 1P03	VCC CPU VID SI0_ 1P03
3	VSS		VSS		DDR 3_0 _MO N2N		DDR 3_0 _CK E[3]		VSS			DDR 3_0 _DQ ECC[7]				DDR 3_0 _DQ ECC[4]		DDR 3_0 _DQ [27]			DDR 3_0 _DQ S[3]				VSS		VCC CPU VID SI0_ 1P03
2											DDR 3_0 _DQ ECC[2]		DDR 3_0 _DQ SEC C[0]		DDR 3_0 _DQ ECC[1]		VSS			DDR 3_0 _DQ [30]		DDR 3_0 _DQ [25]		DDR 3_0 _DQ [28]		VCC CPU VID SI0_ 1P03	
1	VSS		VSS		VSS	VSS		VSS	VSS			DDR 3_0 _DQ ECC[6]			DDR 3_0 _DQ ECC[5]			DDR 3_0 _DQ [26]			DDR 3_0 _DQ SB[3]			VSS			VCC CPU VID SI0_ 1P03

Table 35-5. Top Right (Sheet 1 of 6)

	AG	AF	AE	AD	AC	AB	AA	Y	W	V	U	T	R	P	N	M	L	K	J	H	G	F	E	D	C	B	A
66	VSS			PMU_WAKE_B			VSS			GPIO_SUS0			SATA3_TXP[0]			VSS			VSS	VSS		VSS	VSS		VSS		VSS
65		PMU_SLP_S3_B		TCK		SUS_STAT_B		SPI_CS0_B			VSS		SATA3_TXN[1]		VSS		SATA3_RXN[1]											
64	VSS		VSS				VSS			PMU_SLP_S45_B		SATA3_TXN[0]				SATA3_RXN[0]			VSS		SATA_RXN[1]		VSS		VSS		VSS
63	VSS	TDO		USB_OC0_B		AB63_RSVD		CPU_RESET_B		NCSI_RXD1	VSS		SATA3_TXP[1]		VSS		SATA3_RXP[1]		VSS	SATA_RXN[0]		VSS					
62		VSS	PMU_PLTRST_B	VSS		SPI_MOSI	VSS		VSS		VSS	VSS	VSS		VSS	SATA3_RXP[0]		VSS		SATA_RXP[0]	SATA_RXP[1]		SATA_RXP[2]		SATA_RXN[2]		VSS
61																							VSS				VSS
60	AG60_RSVD							VSS	GBE_EE_DO							VSS	VSS							SATA_RXP[3]	SATA_RXN[3]		
59	LPC_AD3			SPI_MISO	VSS			NCSI_ARB_OUT	GBE_MDIO0_I2C_DATA			GBE_SMBD	GBE_EE_CS_N			VSS	SATA3_REFCLKN			VSS	VSS		VSS	VSS			VSS
58				PMU_SUSCLK	SPI_CS1_B							GBE_SDP0_0	GBE_LED3							VSS			SATA_TXP[1]		VSS		
57	VSS							SUS PWRDNACK	VSS							VSS	SATA3_REFCLKP						SATA_TXN[1]	SATA_TXN[0]		SATA_TXP[0]	VSS
56	FLEX_CLK_SE1			VSS	TDI			VSS	GBE_MDIO0_I2C_CLK			VSS	VSS			VSS				VSS	VSS						
55				BVCCRTC_EXTPAD	VSS							GBE_SMBALRT_N	SATA3_OBSP										VSS	VSS		VSS	
54	LPC_AD0							GBE_MDIO1_I2C_CLK	GPIO_SUS1							VSS	SATA_REFCLKN		SATA_REFCLKP	VSS			VSS		SATA_TXP[2]		SATA_TXN[2]
53	VSS			AD53_RSVD	TMS			GBE_MDIO1_I2C_DATA	VSS			GPIO_SUS2	SATA3_OBSN											SATA_TXP[3]		SATA_TXN[3]	

Table 35-5. Top Right (Sheet 2 of 6)

	AG	AF	AE	AD	AC	AB	AA	Y	W	V	U	T	R	P	N	M	L	K	J	H	G	F	E	D	C	B	A
52				VSS	PMU_SLP_DDRVTT_B							VSS	VSS			VSS	SATA_OBSN		SATA_OBSP	VSS			VSS				
51	LPC_CLKOUT0							VSS	GBE_EE_DI															VSS	VSS		VSS
50	UART1_RXD			RTEST_B	VSS			PMU_SLP_LAN_B	GBE_LED1			GBE_EE_SK		GBE_SMBCLK	VSS		VSS	VSS		GBE_OBSN	GBE_OBSP		VSS	GBE_REFCLKN		GBE_REFCLKP	
49				SRTCRST_B	PMU_PWRBTN_B																		VSS		VSS		
48	VSS							VSS	VSS			GBE_SDP0_1		GBE_LED2	VSS		VSS	GBE_RXN[0]		GBE_RXP[0]	VSS		VSS	GBE_TXP[0]		GBE_TXN[0]	VSS
47			VSS	VSS	RSMRST_B		AA47_RSVD					VSS															
46	RCOMP_CORE_LVT	SPI_CLK		VSS	VSS		TRST_B		VCCAREF_SATA_HVGEN	VCCAREF_SATA_HVGEN			VCCAPLL_SATA_1P0	GBE_LED0		VSS	GBE_RXN[1]		GBE_RXP[1]	VSS			VSS	GBE_TXP[1]		GBE_TXN[1]	
45												VCCAPLL_SATA_1P0											VSS		VSS		VSS
44	VSS	VSS		VCCPADXXXSUS_3P3	VCCFHVSOCSI0_1P03		VSS		VCCAREF_SATA_HVGEN	VSS		VCCAPLL_SATA3_1P0	VCCAPLL_SATA3_1P0	VSS		VSS	GBE_RXN[2]		GBE_RXP[2]	VSS				GBE_TXP[2]		GBE_TXN[2]	
43																							VSS				
42	VCCRTC_3P3	VCCPADXXXSUS_1P8		VCCPADXXXSUS_3P3	VCCFHVSOCSI0_1P03		VSS		VSS	VSS		VSS		VCCA_SATA3_1P0	VCCA_SATA3_1P0		VCCA_SATA3_1P0	VSS		GBE_RXN[3]	GBE_RXP[3]			VSS	GBE_TXP[3]		GBE_TXN[3]
41	VSS	VCCPADXXXSUS_1P8		VSS	VCCDIGXXXSUS_1P03		VCCDIGXXXSUS_1P03		VCCAPLL_GBE_1P0	VCCAPLL_GBE_1P0		VSS		VCCA_SATA_1P0	VCCA_SATA_1P0		VCCA_SATA_1P0	VCCA_SATA_1P0		VSS	VSS		VSS	VSS		VSS	
40																							VSS		DDR3_1_DQ[5]		

Table 35-5. Top Right (Sheet 3 of 6)

	AG	AF	AE	AD	AC	AB	AA	Y	W	V	U	T	R	P	N	M	L	K	J	H	G	F	E	D	C	B	A
39	VNN	VNN		VNN	VCC DIG XXX SUS _1P03		VCC DIG XXX SUS _1P03		VSS	VCC ARE F_G BE_ HVG EN		VCC ARE F_G BE_ HVG EN											VSS	DDR 3_1 _DQ [1]		DDR 3_1 _DQ [0]	DDR 3_1 _DQ [4]
38	VCC DIG XXX SI0_ 1P03	VCC DIG XXX SI0_ 1P03		VCC DIG XXX SI0_ 1P03	VCC DIG XXX SUS _1P03		VCC DIG XXX SUS _1P03		VCC A_G BE_ 1P0	VCC A_G BE_ 1P0			VSS	P38_ RSV D		VSS	L38_ RSV D		J38_ RSV D	VSS							
37												VCC A_G BE_ 1P0	R37 _RS VD	VSS		VSS	HPLL _RE FP		HPLL _RE FN	VSS			VSS	DDR 3_1 _DQ S[0]		DDR 3_1 _DQ SB[0]	
36	VNN	VNN		VNN	VSS		VSS		VSS	VSS		VSS											VSS		DDR 3_1 _DQ [7]		DDR 3_1 _DQ [6]
35																								DDR 3_1 _DQ [3]		DDR 3_1 _DQ [2]	
34	VSS	VSS RAM CPU SI1_ 1P03 _SE NSE		VSS	VSS		VCC SFR XXX SI0_ 1P35		VCC SFR XXX SI0_ 1P35	VCC SFR XXX SI0_ 1P35		VSS		VSS	VSS		VSS	VSS		VSS	VSS		VSS				
33																								VSS	VSS		VSS
32	VCC RAM CPU SI1_ 1P03	VCC RAM CPU SI1_ 1P03 _SE NSE		VCC RAM CPU SI1_ 1P03	VSS		VCC SFR XXX SI0_ 1P35		VSS	VSS		DDR 3_1 _OD TPU		VSS	DDR 3_1 _DQ [26]		DDR 3_1 _DQ [27]	VSS		DDR 3_1 _DQ [28]	VSS		DDR 3_1 _DQ [12]	DDR 3_1 _DQ [13]		DDR 3_1 _DQ [8]	
31	VCC RAM CPU SI1_ 1P03	VCC RAM CPU SI1_ 1P03		VCC RAM CPU SI1_ 1P03	VSS		VSS		VCC PLLD DR_ 1_1P0	VCC SFR PLLD DR_ 1_1P5		VSS											VSS		DDR 3_1 _DQ [9]		
30													VSS	DDR 3_1 _DQ [31]		DDR 3_1 _DQ [30]	VSS		DDR 3_1 _DQ [29]	DDR 3_1 _DQ [24]			VSS	DDR 3_1 _DQ S[1]		DDR 3_1 _DQ SB[1]	VSS
29	VCC RAM CPU SI1_ 1P03	VCC RAM CPU SI1_ 1P03		VCC FHV CPU SI0_ MOD 2_1P03	VSS		VSS		VCC CLK DDR _1_ 1P5	VCC CLK DDR _1_ 1P5		VSS	DDR 3_1 _VR EF	VSS		DDR 3_1 _DQ S[3]	DDR 3_1 _DQ SB[3]		VSS	DDR 3_1 _DQ [25]							
28	VSS	VSS		VCC FHV CPU SI0_ MOD 0_1P03	VSS		VSS		VCC ACK DDR _1_ 1P0	VCC ACK DDR _1_ 1P0		VSS											VSS	DDR 3_1 _DQ [14]		DDR 3_1 _DQ [15]	

Table 35-5. Top Right (Sheet 4 of 6)

	AG	AF	AE	AD	AC	AB	AA	Y	W	V	U	T	R	P	N	M	L	K	J	H	G	F	E	D	C	B	A
27																							VSS		DDR3_1_DQ[11]		DDR3_1_DQ[10]
26	VSS	VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	AC26_RSVD		VSS		VCCADDR_1_1P0	VCCADLLDDR_1_1P0		DDR3_1_DQPU		VCCDDR_1_1P5	DDR3_1_CKE[0]		DDR3_1_CKE[3]	VCCDDR_1_1P5		DDR3_1_MA[15]	DDR3_1_MA[14]			VSS		VSS	
25	VSS	VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	AC25_RSVD		VSS		VCCADDR_1_1P0	VCCADLLDDR_1_1P0		DDR3_1_CMDPU		VCCDDR_1_1P5	DDR3_1_CKE[1]		DDR3_1_CKE[2]	VCCDDR_1_1P5		DDR3_1_BS[2]	DDR3_1_MA[12]		VSS				
24																								DDR3_1_DQ[21]	DDR3_1_DQ[20]		VSS
23	VSS	VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	VSS		VSS		VCCADDR_1_1P0	VCCADLLDDR_1_1P0		VSS											VSS	DDR3_1_DQ[16]		DDR3_1_DQ[17]	
22													VCCDDR_1_1P5	DDR3_1_MA[11]		DDR3_1_MA[9]	VCCDDR_1_1P5		DDR3_1_MA[7]	DDR3_1_MA[5]			VSS		DDR3_1_DQSB[2]		
21	VCCCPUVIDSI0_1P03	VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	VSS		VCCADDR_1_1P0		VCCADDR_1_1P0	VCCADLLDDR_1_1P0		VSS	VCCDDR_1_1P5	DDR3_1_MA[3]		DDR3_1_MA[4]	VCCDDR_1_1P5		DDR3_1_MA[6]	DDR3_1_MA[8]			VSS	DDR3_1_DQ[23]		DDR3_1_DQ[22]	DDR3_1_DQS[2]
20		VCCCPUVIDSI0_1P03		VSS	VSS	VSS				VCCADLLDDR_1_1P0		VSS															
19	VCCCPUVIDSI0_1P03							VSS	VSS														VSS	DDR3_1_DQ[18]		DDR3_1_DQ[19]	
18				VSS	VSS							VSS		VCCDDR_1_1P5	DDR3_1_BS[0]		DDR3_1_MA[10]	VCCDDR_1_1P5		DDR3_1_BS[1]	DDR3_1_MA[2]		VSS		VSS		VSS
17	VCCCPUVIDSI0_1P03			VSS	DDR3_1_DQ[59]			VSS	VSS			DDR3_1_DQ[36]		VCCDDR_1_1P5	DDR3_1_VCCA_PWROK		DDR3_1_DRAM_PWROK	VCCDDR_1_1P5		DDR3_1_MA[0]	DDR3_1_MA[1]			DDR3_1_DQECC[4]		DDR3_1_DQECC[5]	
16	VSS							DDR3_1_DQ[52]	VSS														VSS				
15				DDR3_1_DQ[58]	DDR3_1_DQ[63]						DDR3_1_DQ[37]		DDR3_1_DQ[32]											DDR3_1_DQECC[0]	DDR3_1_DQECC[1]		VSS

Table 35-5. Top Right (Sheet 5 of 6)

	AG	AF	AE	AD	AC	AB	AA	Y	W	V	U	T	R	P	N	M	L	K	J	H	G	F	E	D	C	B	A
14	VCC CPU VID SI0_ 1P03			VSS	DDR 3_1 _DQ [62]			DDR 3_1 _DQ [53]	DDR 3_1 _DQ [48]		VSS		DDR 3_1 _DQ [33]		VCC DDR _1_ 1P5		DDR 3_1 _RE FN		DDR 3_1 _RE FP	VCC DDR _1_ 1P5			VSS	DDR 3_1 _DQ SBE CC[0]		DDR 3_1 _DQ SEC C[0]	
13	VCC CPU VID SI0_ 1P03							VSS	DDR 3_1 _DQ [49]						VCC DDR _1_ 1P5		DDR 3_1 _CK B[3]		DDR 3_1 _CK[3]	VCC DDR _1_ 1P5			VSS		DDR 3_1 _DQ ECC[6]		
12				DDR 3_1 _DQ S[7]	VSS						DDR 3_1 _DQ SB[4]		VSS										DDR 3_1 _DQ ECC[3]	DDR 3_1 _DQ ECC[2]		DDR 3_1 _DQ ECC[7]	VSS
11	VCC CPU VID SI0_ 1P03			DDR 3_1 _DQ SB[7]	DDR 3_1 _DQ [60]			DDR 3_1 _DQ [51]	VSS		DDR 3_1 _DQ S[4]		DDR 3_1 _DQ [35]		DDR 3_1 _DR AMR STB												
10	VSS							DDR 3_1 _DQ [55]	DDR 3_1 _DQ [50]						VSS			VCC DDR _1_ 1P5		DDR 3_1 _CK[2]	DDR 3_1 _CK B[2]		VSS	VSS		VSS	
9				VSS	DDR 3_1 _DQ [61]						VSS		DDR 3_1 _DQ [34]							DDR 3_1 _CK B[1]	DDR 3_1 _CK[1]		VCC DDR _1_ 1P5		DDR 3_1 _CK[0]		VSS
8	VCC CPU VID SI0_ 1P03			DDR 3_1 _DQ [57]	DDR 3_1 _DQ [56]			VSS	DDR 3_1 _DQ [54]		DDR 3_1 _DQ [39]		DDR 3_1 _DQ [38]		DDR 3_1 _OD T[2]		DDR 3_1 _RA SB							DDR 3_1 _CK B[0]			DDR 3_1 _MO N2P
7	VCC CPU VID SI0_ 1P03							DDR 3_1 _DQ S[6]	DDR 3_1 _DQ SB[6]						DDR 3_1 _WE B		DDR 3_1 _CS B[2]				VCC DDR _1_ 1P5		VCC DDR _1_ 1P5					
6																								VSS	VSS		DDR 3_1 _MO N2N
5	VSS		VCC CPU VID SI0_ 1P03	VSS		VSS	VSS	VSS		VSS		VSS	VSS		VSS	VSS	VCC DDR _1_ 1P5		VCC DDR _1_ 1P5		VCC DDR _1_ 1P5	VSS	DDR 3_1 _MO N1P		DDR 3_1 _MO N1N		VSS
4		VCC CPU VID SI0_ 1P03		VSS		DDR 3_1 _DQ [43]		DDR 3_1 _DQ [46]	VSS		DDR 3_1 _DQ [40]		VSS		DDR 3_1 _CS B[1]		DDR 3_1 _OD T[3]	DDR 3_1 _CS B[3]		DDR 3_1 _CA SB		VSS					

Table 35-5. Top Right (Sheet 6 of 6)

	AG	AF	AE	AD	AC	AB	AA	Y	W	V	U	T	R	P	N	M	L	K	J	H	G	F	E	D	C	B	A
3			VCC CPU VID SI0_ 1P03 _SE NSE				VSS		DDR 3_1 _DQ SB[5]			DDR 3_1 _DQ [44]				VCC DDR _1_ 1P5		DDR 3_1 _CS B[0]			DDR 3_1 _MA [13]		VSS		VSS		
2		VSS CPU VID SI0_ 1P03 _SE NSE		VSS		DDR 3_1 _DQ [42]		DDR 3_1 _DQ [47]			DDR 3_1 _DQ [41]		VSS		DDR 3_1 _OD T[1]		DDR 3_1 _OD T[0]										
1			VCC CPU VID SI0_ 1P03			VSS			DDR 3_1 _DQ S[5]			DDR 3_1 _DQ [45]			VSS			VSS		VSS		VSS	VSS				

Table 35-6. Bottom Left (Sheet 1 of 6)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y	AA	AB	AC	AD	AE	AF	AG
66	VSS		VSS		VSS	VSS		VSS	VSS			VSS			SATA3_TXP[0]			GPIO_SUS0			VSS			PMU_WAKE_B			VSS
65											SATA3_RXN[1]		VSS		SATA3_TXN[1]		VSS			SPI_CS0_B		SUS_STAT_B		TCK		PMU_SLP_S3_B	
64	VSS		VSS		VSS		SATA_RXN[1]		VSS			SATA3_RXN[0]				SATA3_TXN[0]		PMU_SLP_S45_B			VSS				VSS		VSS
63						VSS		SATA_RXN[0]	VSS		SATA3_RXP[1]		VSS		SATA3_TXP[1]		VSS	NCSI_RXD1		CPU_RESET_B		AB63_RSVD		USB_OC0_B		TDO	VSS
62	VSS		SATA_RXN[2]		SATA_RXP[2]		SATA_RXP[1]	SATA_RXP[0]		VSS		SATA3_RXP[0]	VSS		VSS	VSS	VSS		VSS		VSS	SPI_MOSI		VSS	PMU_PLTRST_B	VSS	
61	VSS				VSS																						
60			SATA_RXN[3]	SATA_RXP[3]							VSS	VSS							GBE_EE_DO	VSS							AG60_RSVD
59	VSS			VSS	VSS		VSS	VSS			SATA3_REFCLKN	VSS			GBE_EE_CS_N	GBE_SMBD			GBE_MDIO0_I2C_DATA	NCSI_ARB_OUT			VSS	SPI_MISO			LPC_AD3
58			VSS		SATA_TXP[1]			VSS							GBE_LED3	GBE_SDP0_0							SPI_CS1_B	PMU_SUSCLK			
57	VSS	SATA_TXP[0]		SATA_TXN[0]	SATA_TXN[1]						SATA3_REFCLKP	VSS							VSS	SUSPWRDNACK							VSS
56							VSS	VSS				VSS			VSS	VSS			GBE_MDIO0_I2C_CLK	VSS			TDI	VSS			FLEX_CLK_SE1
55		VSS		VSS	VSS										SATA3_OBSP	GBE_SMBALRT_N							VSS	BVCCRTC_EXTPAD			
54	SATA_TXN[2]		SATA_TXP[2]		VSS			VSS	SATA_REFCLKP		SATA_REFCLKN	VSS							GPIO_SUS1	GBE_MDIO1_I2C_CLK							LPC_AD0
53		SATA_TXN[3]		SATA_TXP[3]											SATA3_OBSN	GPIO_SUS2			VSS	GBE_MDIO1_I2C_DATA			TMS	INTRUDER_B			VSS

Table 35-6. Bottom Left (Sheet 2 of 6)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y	AA	AB	AC	AD	AE	AF	AG
52					VSS			VSS	SATA_OBSP		SATA_OBSN	VSS			VSS	VSS							PMU_SLP_DDRVTT_B	VSS			
51	VSS		VSS	VSS															GBE_EE_DI	VSS							LPC_CLKOUT0
50		GBE_REFCLKP		GBE_REFCLKN	VSS		GBE_OBSP	GBE_OBSN		VSS	VSS		VSS	GBE_SMBCLK		GBE_EE_SK			GBE_LED1	PMU_SLP_LAN_B			VSS	RTEST_B			UART1_RXD
49			VSS		VSS																		PMU_PWRBTN_B	SRTCRST_B			
48	VSS	GBE_TXN[0]		GBE_TXP[0]	VSS		VSS	GBE_RXP[0]		GBE_RXN[0]	VSS		VSS	GBE_LED2		GBE_SDP0_1			VSS	VSS							VSS
47																VSS					PMU_ACPRESENT		RSMRST_B	VSS	VSS		
46		GBE_TXN[1]		GBE_TXP[1]	VSS			VSS	GBE_RXP[1]		GBE_RXN[1]	VSS		GBE_LED0	VCCAPLLSATA_1P0			VCCAREF_SATA_HVGEN	VCCAREF_SATA_HVGEN		TRST_B		VSS	VSS		SPI_CLK	RCOMP_CORE_LVT
45	VSS		VSS		VSS											VCCAPLL_SATA_1P0											
44		GBE_TXN[2]		GBE_TXP[2]				VSS	GBE_RXP[2]		GBE_RXN[2]	VSS		VSS	VCCAPLL_SATA3_1P0	VCCAPLL_SATA3_1P0		VSS	VCCAREF_SATA_HVGEN		VSS		VCCFHVSOCSI0_1P03	VCCPADXXXSUS_3P3		VSS	VSS
43					VSS																						
42	GBE_TXN[3]		GBE_TXP[3]	VSS			GBE_RXP[3]	GBE_RXN[3]		VSS	VCCA_SATA3_1P0		VCCA_SATA3_1P0	VCCA_SATA3_1P0		VSS		VSS	VSS		VSS		VCCFHVSOCSI0_1P03	VCCPADXXXSUS_3P3		VCCPADXXXSUS_1P8	VCCRTC_3P3
41		VSS		VSS	VSS		VSS	VSS		VCCA_SATA_1P0	VCCA_SATA_1P0		VCCA_SATA_1P0	VCCA_SATA_1P0		VSS		VCCAPLL_GBE_1P0	VCCAPLL_GBE_1P0		VCCDIGXXXSUS_1P03		VCCDIGXXXSUS_1P03	VSS		VCCPADXXXSUS_1P8	VSS
40			DDR3_1_DQ[5]		VSS																						

Table 35-6. Bottom Left (Sheet 3 of 6)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y	AA	AB	AC	AD	AE	AF	AG
39	DDR3_1_DQ[4]	DDR3_1_DQ[0]		DDR3_1_DQ[1]	VSS											VCCAREF_GBE_HVGEN		VCCAREF_GBE_HVGEN	VSS		VCCDIGXXXSUS_1P03		VCCDIGXXXSUS_1P03	VNN		VNN	VNN
38								VSS	J38_RSVD		L38_RSVD	VSS		P38_RSVD	VSS			VCCA_GBE_1P0	VCCA_GBE_1P0		VCCDIGXXXSUS_1P03		VCCDIGXXXSUS_1P03	VCCDIGXXXSI0_1P03		VCCDIGXXXSI0_1P03	VCCDIGXXXSI0_1P03
37		DDR3_1_DQSB[0]		DDR3_1_DQS[0]	VSS			VSS	HPLL_REFN		HPLL_REFP	VSS		VSS	R37_RSVD	VCCA_GBE_1P0											
36	DDR3_1_DQ[6]		DDR3_1_DQ[7]		VSS											VSS		VSS	VSS		VSS		VSS	VNN		VNN	VNN
35		DDR3_1_DQ[2]		DDR3_1_DQ[3]																							
34					VSS		VSS	VSS		VSS	VSS		VSS	VSS		VSS		VCCSFRXXXSI0_1P35	VCCSFRXXXSI0_1P35		VCCSFRXXXSI0_1P35		VSS	VSS		VSSRAMCPUSI1_1P03_SENSE	VSS
33	VSS		VSS	VSS																							
32		DDR3_1_DQ[8]		DDR3_1_DQ[13]	DDR3_1_DQ[12]		VSS	DDR3_1_DQ[28]		VSS	DDR3_1_DQ[27]		DDR3_1_DQ[26]	VSS		DDR3_1_ODTPU		VSS	VSS		VCCSFRXXXSI0_1P35		VSS	VCCRAMCPUSI1_1P03		VCCRAMCPUSI1_1P03_SENSE	VCCRAMCPUSI1_1P03
31			DDR3_1_DQ[9]		VSS											VSS		VCCSFRPLLDDR_1_1P5	VCCPLLDDR_1_1P0		VSS		VSS	VCCRAMCPUSI1_1P03		VCCRAMCPUSI1_1P03	VCCRAMCPUSI1_1P03
30	VSS	DDR3_1_DQSB[1]		DDR3_1_DQS[1]	VSS			DDR3_1_DQ[24]	DDR3_1_DQ[29]		VSS	DDR3_1_DQ[30]		DDR3_1_DQ[31]	VSS												
29								DDR3_1_DQ[25]	VSS		DDR3_1_DQSB[3]	DDR3_1_DQS[3]		VSS	DDR3_1_VREF	VSS		VCCCLKDDR_1_1P5	VCCCLKDDR_1_1P5		VSS		VSS	VCCFHVCPUSI0_MOD2_1P03		VCCRAMCPUSI1_1P03	VCCRAMCPUSI1_1P03
28		DDR3_1_DQ[15]		DDR3_1_DQ[14]	VSS											VSS		VCCACKDDR_1_1P0	VCCACKDDR_1_1P0		VSS		VSS	VCCFHVCPUSI0_MOD0_1P03		VSS	VSS

Table 35-6. Bottom Left (Sheet 4 of 6)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y	AA	AB	AC	AD	AE	AF	AG
27	DDR3_1_DQ[10]		DDR3_1_DQ[11]		VSS																						
26		VSS		VSS			DDR3_1_MA[14]	DDR3_1_MA[15]		VCCDDR_1_1P5	DDR3_1_CKE[3]		DDR3_1_CKE[0]	VCCDDR_1_1P5		DDR3_1_DQPU		VCCADLLDDR_1_1P0	VCCADDR_1_1P0		VSS		AC26_RSVD	VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	VSS
25					VSS		DDR3_1_MA[12]	DDR3_1_BS[2]		VCCDDR_1_1P5	DDR3_1_CKE[2]		DDR3_1_CKE[1]	VCCDDR_1_1P5		DDR3_1_CMDPU		VCCADLLDDR_1_1P0	VCCADDR_1_1P0		VSS		AC25_RSVD	VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	VSS
24	VSS		DDR3_1_DQ[20]	DDR3_1_DQ[21]																							
23		DDR3_1_DQ[17]		DDR3_1_DQ[16]	VSS											VSS		VCCADLLDDR_1_1P0	VCCADDR_1_1P0		VSS		VSS	VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	VSS
22			DDR3_1_DQSB[2]		VSS			DDR3_1_MA[5]	DDR3_1_MA[7]		VCCDDR_1_1P5	DDR3_1_MA[9]		DDR3_1_MA[11]	VCCDDR_1_1P5												
21	DDR3_1_DQS[2]	DDR3_1_DQ[22]		DDR3_1_DQ[23]	VSS			DDR3_1_MA[8]	DDR3_1_MA[6]		VCCDDR_1_1P5	DDR3_1_MA[4]		DDR3_1_MA[3]	VCCDDR_1_1P5	VSS		VCCADLLDDR_1_1P0	VCCADDR_1_1P0		VCCADDR_1_1P0		VSS	VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	VCCCPUVIDSI0_1P03
20																VSS		VCCADLLDDR_1_1P0				VSS	VSS	VSS		VCCCPUVIDSI0_1P03	
19		DDR3_1_DQ[19]		DDR3_1_DQ[18]	VSS														VSS	VSS							VCCCPUVIDSI0_1P03
18	VSS		VSS		VSS		DDR3_1_MA[2]	DDR3_1_BS[1]		VCCDDR_1_1P5	DDR3_1_MA[10]		DDR3_1_BS[0]	VCCDDR_1_1P5		VSS							VSS	VSS			
17		DDR3_1_DQECC[5]		DDR3_1_DQECC[4]			DDR3_1_MA[1]	DDR3_1_MA[0]		VCCDDR_1_1P5	DDR3_1_DRAM_PWROK		DDR3_1_VCCA_PWROK	VCCDDR_1_1P5		DDR3_1_DQ[36]			VSS	VSS				DDR3_1_DQ[59]	VSS			VCCCPUVIDSI0_1P03
16					VSS														VSS	DDR3_1_DQ[52]							VSS
15	VSS		DDR3_1_DQECC[1]	DDR3_1_DQECC[0]											DDR3_1_DQ[32]		DDR3_1_DQ[37]						DDR3_1_DQ[63]	DDR3_1_DQ[58]			

Table 35-6. Bottom Left (Sheet 5 of 6)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y	AA	AB	AC	AD	AE	AF	AG
14		DDR3_1_DQSECC[0]		DDR3_1_DQSBECC[0]	VSS			VCCDDR_1_1P5	DDR3_1_REFP		DDR3_1_REFN		VCCDDR_1_1P5		DDR3_1_DQ[33]		VSS		DDR3_1_DQ[48]	DDR3_1_DQ[53]			DDR3_1_DQ[62]	VSS			VCCCPUVIDSI0_1P03
13			DDR3_1_DQECC[6]		VSS			VCCDDR_1_1P5	DDR3_1_CK[3]		DDR3_1_CKB[3]		VCCDDR_1_1P5						DDR3_1_DQ[49]	VSS							VCCCPUVIDSI0_1P03
12	VSS	DDR3_1_DQECC[7]		DDR3_1_DQECC[2]	DDR3_1_DQECC[3]										VSS		DDR3_1_DQSB[4]						VSS	DDR3_1_DQS[7]			
11													DDR3_1_DRAMRSTB		DDR3_1_DQ[35]		DDR3_1_DQS[4]		VSS	DDR3_1_DQ[51]			DDR3_1_DQ[60]	DDR3_1_DQSB[7]			VCCCPUVIDSI0_1P03
10		VSS		VSS	VSS		DDR3_1_CKB[2]	DDR3_1_CK[2]		VCCDDR_1_1P5			VSS						DDR3_1_DQ[50]	DDR3_1_DQ[55]							VSS
9	VSS		DDR3_1_CK[0]		VCCDDR_1_1P5		DDR3_1_CK[1]	DDR3_1_CKB[1]							DDR3_1_DQ[34]		VSS						DDR3_1_DQ[61]	VSS			
8	DDR3_1_MON2P			DDR3_1_CKB[0]							DDR3_1_RASB		DDR3_1_ODT[2]		DDR3_1_DQ[38]		DDR3_1_DQ[39]		DDR3_1_DQ[54]	VSS			DDR3_1_DQ[56]	DDR3_1_DQ[57]			VCCCPUVIDSI0_1P03
7					VCCDDR_1_1P5		VCCDDR_1_1P5				DDR3_1_CSB[2]		DDR3_1_WEB						DDR3_1_DQSB[6]	DDR3_1_DQS[6]							VCCCPUVIDSI0_1P03
6	DDR3_1_MON2N		VSS	VSS																							
5	VSS		DDR3_1_MON1N		DDR3_1_MON1P	VSS	VCCDDR_1_1P5		VCCDDR_1_1P5		VCCDDR_1_1P5	VSS	VSS		VSS	VSS		VSS		VSS	VSS	VSS		VSS	VCCCPUVIDSI0_1P03		VSS
4						VSS		DDR3_1_CASB		DDR3_1_CSB[3]	DDR3_1_ODT[3]		DDR3_1_CSB[1]		VSS		DDR3_1_DQ[40]		VSS	DDR3_1_DQ[46]		DDR3_1_DQ[43]		VSS		VCCCPUVIDSI0_1P03	

Table 35-6. Bottom Left (Sheet 6 of 6)

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R	T	U	V	W	Y	AA	AB	AC	AD	AE	AF	AG
3			VSS		VSS		DDR3_1_MA[13]			DDR3_1_CSB[0]		VCCDDR_1_1P5				DDR3_1_DQ[44]			DDR3_1_DQSB[5]		VSS				VCCCPUVIDSI0_1P03_SENSE		
2											DDR3_1_ODT[0]		DDR3_1_ODT[1]		VSS		DDR3_1_DQ[41]			DDR3_1_DQ[47]		DDR3_1_DQ[42]		VSS		VSSCPUVIDSI0_1P03_SENSE	
1					VSS	VSS		VSS		VSS			VSS			DDR3_1_DQ[45]			DDR3_1_DQS[5]			VSS			VCCCPUVIDSI0_1P03		

Table 35-7. Bottom Right (Sheet 1 of 5)

	AH	AJ	AK	AL	AM	AN	AP	AR	AT	AU	AV	AW	AY	BA	BB	BC	BD	BE	BF	BG	BH	BJ	BK	BL	BM	BN	BP
66			VSS			VSS			VSS			DFX_PORT13			VSS			VSS		VSS		VSS	VSS		VSS		VSS
65		BRTCX1_PAD		ERROR0_B		SMB_DATA2		SMB_CLK2			DFX_PORT_CLK1		DFX_PORT14		VSS		VSS										
64			VSS				VSS		VSS			DFX_PORT12				DFX_PORT0		PCIE_RXN[0]			PCIE_RXN[1]		VSS		VSS		VSS
63		BRTCX2_PAD		ERROR2_B		SMB_DATA1		SMB_CLK1	SATA_GP0		DFX_PORT_CLK0		DFX_PORT6		DFX_PORT10		VSS	PCIE_RXP[0]		VSS		PCIE_RXP[1]					
62	VSS		VSS	ERROR1_B		SMB_CLK0	SMB_DATA0	VSS		VSS		DFX_PORT7	VSS		VSS	SVID_ALERT_B	VSS		VSS		VSS	PCIE_RXP[2]	PCIE_RXN[2]		VSS		VSS
61																								PCIE_TXN[0]	VSS		VSS
60	COREPWROK							VSS	DFX_PORT9						SVID_CLK		VSS				VSS		PCIE_TXP[0]				
59	FLEX_CLK_SE0			VSS	VSS			DFX_PORT15	DFX_PORT11			VSS	DFX_PORT8		PROCHOT_B		VSS							PCIE_TXP[1]			VSS
58				SMBALRT_N0	PMU_RESETBUTTON_B							DFX_PORT2	DFX_PORT1							PCIE_RXP[3]	PCIE_RXN[3]		VSS		PCIE_TXN[1]		VSS
57	VSS							DFX_PORT3	VSS						VSS			VSS		PCIE_RXP[4]	PCIE_RXN[4]		VSS	PCIE_TXP[2]		PCIE_TXN[2]	
56	LPC_FRAMEB			NMI	CLK14_IN			VSS	THERMTRIP_N			CX_PRDY_B	DFX_PORT5		VSS								VSS				
55				VSS	VSS							VSS	VSS											PCIE_TXP[3]		PCIE_TXN[3]	VSS
54	SATA3_LEDN							AR54_RSVD	DFX_PORT4						VSS		PCIE_RXP[5]		PCIE_RXN[5]	VSS			PCIE_TXN[4]		PCIE_TXP[4]		
53	VSS			LPC_AD2	LPC_AD1			AR53_RSVD	VSS			SVID_DATA	CX_PREQ_B		VSS		VSS		PCIE_RXP[6]	PCIE_RXN[6]			VSS	PCIE_TXN[5]		PCIE_TXP[5]	
52				MCERR_B	IERR_B							VSS	VSS											PCIE_TXP[6]	PCIE_TXN[6]		VSS
51	SATA3_GP0							AR51_RSVD	AT51_RSVD														VSS				

Table 35-7. Bottom Right (Sheet 2 of 5)

	AH	AJ	AK	AL	AM	AN	AP	AR	AT	AU	AV	AW	AY	BA	BB	BC	BD	BE	BF	BG	BH	BJ	BK	BL	BM	BN	BP
50	UART1_TXD			VSS	VSS			VSS	ILB_SERIRQ			VSS		USB_RCOMP_FCLKN	USB_RCOMP_FCLKP		VSS	PCIE_RXP[7]		PCIE_RXN[7]	VSS			PCIE_TXN[7]		PCIE_TXP[7]	
49				SATA_LEDN	LPC_CLKOUT1							VSS		VSS	PCIE_RCOMP_FCLKP		PCIE_RCOMP_FCLKN	VSS		PCIE_RXP[8]	PCIE_RXN[8]		VSS		PCIE_TXN[8]		PCIE_TXP[8]
48	LPC_CLKRUNB							USB_OBSP	USB_RCOMPO														VSS	PCIE_TXN[9]		PCIE_TXP[9]	
47		VSS		CTB TRIGOUT	CTB TRIGINOUT	VSS				VCCUSBSUS_1P8		VSS															
46		VSS		MEMHOT_B	VSS		VSS		USB_RCOMPI	VCCUSBSUS_1P8		VSS	USB_DP[0]	USB_DN[0]		VSS	PCIE_RXP[9]		PCIE_RXN[9]	VSS			VSS	PCIE_TXN[10]		PCIE_TXP[10]	VSS
45													VSS	USB_DP[1]		USB_DN[1]	VSS		PCIE_RXP[10]	PCIE_RXN[10]			VSS		VSS		
44		VSS		VSS	VSS		VSS		VSS	VSSA_USB		VSSA_USB											VSS	PCIE_TXN[11]		PCIE_TXP[11]	
43																								VSS	PCIE_TXN[12]		PCIE_TXP[12]
42		VCCPADXXXSI0_3P3		VSS	VNN		VSS		VCCAUSB_1P0	VCCUSBSUS_3P3		VCCUSBSUS_3P3		VSS	USB_DN[2]		USB_DP[2]	VSS		PCIE_RXP[11]	PCIE_RXN[11]		VSS				
41		VCCPADXXXSI0_1P8		VCCPADXXXSI0_1P8	VNN		VSS		VCCDUSBSUS_1P0	VCCDUSB_1P0		VCCDUSB_1P0		USB_DN[3]	USB_DP[3]		VSS	PCIE_RXP[12]		PCIE_RXN[12]	VSS			PCIE_TXP[13]		PCIE_TXN[13]	
40																							PCIE_TXN[14]		PCIE_TXP[14]		VSS
39		VNN		VNN	VNN		VSS		VCCAREF_PCIE_HVGEN	VCCA_PCIE_1P0		VCCA_PCIE_1P0											VSS	PCIE_TXN[15]		PCIE_TXP[15]	
38		VCCDIGXXXSI0_1P03		VSS	VNN		VSS		VCCA_PCIE_1P0	VCCA_PCIE_1P0		VCCA_PCIE_1P0	VSS	PCIE_OBSN		PCIE_OBSP	VSS		PCIE_RXP[13]	PCIE_RXN[13]							
37													VCCAPLL_PCIE_1P0	VCCAPLL_PCIE_1P0		VSS	PCIE_RXP[14]		PCIE_RXN[14]	VSS			VSS	VSS		VSS	VSS

Table 35-7. Bottom Right (Sheet 3 of 5)

	AH	AJ	AK	AL	AM	AN	AP	AR	AT	AU	AV	AW	AY	BA	BB	BC	BD	BE	BF	BG	BH	BJ	BK	BL	BM	BN	BP
36		VNN		VNN	VNN		VSS		VCCA_PCIE_1P0	VCCA_PCIE_1P0		VCCA_PCIE_1P0											VSS		DDR3_0_DQ[59]		
35																							DDR3_0_DQ[62]	DDR3_0_DQ[63]		DDR3_0_DQ[58]	
34		AJ34_RSVD		AL34_RSVD	VNN		VSS		AT34_RSVD	AU34_RSVD		VSS		VSS	VSS		VSS	PCIE_RXP[15]		PCIE_RXN[15]	VSS			VSS	DDR3_0_DQSB[7]		DDR3_0_DQS[7]
33																											
32		VSS		VCCCORE6VIDSI0GT_1P03	VNN		VSS		VCCACKDDR_0_1P0	VCCACKDDR_0_1P0		VSS		DDR3_0_DQ[32]	DDR3_0_DQ[36]		VSS	VSS		VSS	VSS		DDR3_0_DQ[57]	DDR3_0_DQ[61]		DDR3_0_DQ[56]	
31		VCCRAMCPUSI0GT_MOD3_1P03		VCCCORE7VIDSI0GT_1P03	VNN		VSS		VCCSFRPLLDDR_0_1P5	VCCPLLDDR_0_1P0		VSS											VSS		DDR3_0_DQ[60]		VSS
30												VSS	DDR3_0_DQ[37]	DDR3_0_DQ[33]		VSS	DDR3_0_DQ[35]		DDR3_0_DQ[34]	DDR3_0_DQ[39]			VSS	VSS		VSS	
29		VCCFHVCPUSI0_MOD3_1P03		VSS	VNN		VSS		VCCCLKDDR_0_1P5	VCCCLKDDR_0_1P5			DDR3_0_VREF	VSS		DDR3_0_DQSB[4]	DDR3_0_DQS[4]		VSS	DDR3_0_DQ[38]							
28		VSS		VCCFHVCPUSI0_MOD1_1P03	VNN		VSS		VCCADDR_0_1P0	VCCADLLDDR_0_1P0		VSS											VSS	DDR3_0_DQ[50]		DDR3_0_DQ[51]	DDR3_0_DQ[55]
27																							VSS		DDR3_0_DQ[54]		
26		VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	VNN		VSS		VCCADDR_0_1P0	VCCADLLDDR_0_1P0		VSS		DDR3_0_CMDPU	DDR3_0_ODT[1]		DDR3_0_ODT[0]	VCCDDR_0_1P5		DDR3_0_CSB[3]	DDR3_0_CASB		VSS	DDR3_0_DQSB[6]		DDR3_0_DQS[6]	
25		VCCCPUVIDSI0_1P03		VCCCPUVIDSI0_1P03	VNN		VSS		VCCADDR_0_1P0	VCCADLLDDR_0_1P0		VSS		DDR3_0_DRAMRSTB	DDR3_0_CSB[1]		VCCDDR_0_1P5	DDR3_0_CSB[0]		DDR3_0_ODT[3]	DDR3_0_ODT[2]			VSS	DDR3_0_DQ[48]		DDR3_0_DQ[49]
24																							VSS				

Table 35-7. Bottom Right (Sheet 4 of 5)

	AH	AJ	AK	AL	AM	AN	AP	AR	AT	AU	AV	AW	AY	BA	BB	BC	BD	BE	BF	BG	BH	BJ	BK	BL	BM	BN	BP
23		VCC CPU VID SI0_ 1P03		VCC CPU VID SI0_ 1P03	VSS		VSS		VCC ADD R_0 _1P0	VCC ADL LDD R_0 _1P0		VSS	VCC DDR _0_ 1P5	DDR 3_0 _MA [13]		DDR 3_0 _WE B	DDR 3_0 _CS B[2]		VCC DDR _0_ 1P5	DDR 3_0 _RA SB				DDR 3_0 _DQ [52]		DDR 3_0 _DQ [53]	
22																							VSS		VSS		VSS
21		VCC CPU VID SI0_ 1P03		VCC CPU VID SI0_ 1P03	VSS		AP2 1_R SVD		VCC ADD R_0 _1P0	VCC ADL LDD R_0 _1P0		DDR 3_0 _DQ PU	VCC DDR _0_ 1P5	DDR 3_0 _CK[0]		DDR 3_0 _CK B[0]	VCC DDR _0_ 1P5		DDR 3_0 _CK[1]	DDR 3_0 _CK B[1]			VSS	DDR 3_0 _DQ [42]		DDR 3_0 _DQ [43]	
20			VCC CPU VID SI0_ 1P03	VSS	VSS		AP2 0_R SVD					DDR 3_0 _OD TPU															
19	VCC CPU VID SI0_ 1P03							VSS	VSS			VSS	VCC DDR _0_ 1P5		DDR 3_0 _CK[3]		DDR 3_0 _CK B[3]	VCC DDR _0_ 1P5		DDR 3_0 _CK[2]	DDR 3_0 _CK B[2]		VSS	VSS		DDR 3_0 _DQ [47]	DDR 3_0 _DQ [46]
18				VSS	VSS																		VSS		DDR 3_0 _DQ S[5]		
17	VSS			VSS	DDR 3_0 _DQ [4]			VSS	VSS			DDR 3_0 _DQ [19]		VSS	VSS		VCC DDR _0_ 1P5	DDR 3_0 _VC CA_ PWR OK		DDR 3_0 _DR AM_ PWR OK	VCC DDR _0_ 1P5		VSS	DDR 3_0 _DQ [41]		DDR 3_0 _DQ SB[5]	
16	VCC CPU VID SI0_ 1P03							DDR 3_0 _DQ [12]	VSS															DDR 3_0 _DQ [40]	DDR 3_0 _DQ [44]		DDR 3_0 _DQ [45]
15				DDR 3_0 _DQ [5]	DDR 3_0 _DQ [0]							DDR 3_0 _DQ [23]	DDR 3_0 _DQ [18]			VCC DDR _0_ 1P5	DDR 3_0 _MA [9]		DDR 3_0 _MA [15]	VCC DDR _0_ 1P5			VSS				
14	VCC CPU VID SI0_ 1P03			VSS	DDR 3_0 _DQ [1]			DDR 3_0 _DQ [8]	DDR 3_0 _DQ [13]			VSS	DDR 3_0 _DQ [22]											VSS		VSS	
13	VCC CPU VID SI0_ 1P03							VSS	DDR 3_0 _DQ [9]							VCC DDR _0_ 1P5	DDR 3_0 _MA [4]		VCC DDR _0_ 1P5	DDR 3_0 _MA [2]			VCC DDR _0_ 1P5		DDR 3_0 _RE FN		VSS
12				DDR 3_0 _DQ SB[0]	VSS							DDR 3_0 _DQ S[2]	VSS										VCC DDR _0_ 1P5	DDR 3_0 _BS[0]		DDR 3_0 _RE FP	
11	VSS			DDR 3_0 _DQ S[0]	DDR 3_0 _DQ [6]			DDR 3_0 _DQ [11]	VSS			DDR 3_0 _DQ SB[2]	DDR 3_0 _DQ [17]			DDR 3_0 _MA [14]				DDR 3_0 _MA [3]	DDR 3_0 _MA [0]						
10	VCC CPU VID SI0_ 1P03							DDR 3_0 _DQ [15]	DDR 3_0 _DQ [10]							VCC DDR _0_ 1P5	DDR 3_0 _MA [6]						VSS	DDR 3_0 _BS[1]		DDR 3_0 _MA [10]	VSS

Table 35-7. Bottom Right (Sheet 5 of 5)

	AH	AJ	AK	AL	AM	AN	AP	AR	AT	AU	AV	AW	AY	BA	BB	BC	BD	BE	BF	BG	BH	BJ	BK	BL	BM	BN	BP
9				VSS	DDR3_0_DQ[7]							VSS	DDR3_0_DQ[16]							VCCDDR_0_1P5			VCCDDR_0_1P5		VCCDDR_0_1P5		
8	VCCCPUVIDSI0_1P03			DDR3_0_DQ[2]	DDR3_0_DQ[3]			VSS	DDR3_0_DQ[14]			DDR3_0_DQ[20]	DDR3_0_DQ[21]			DDR3_0_BS[2]	DDR3_0_MA[8]			DDR3_0_MA[11]	DDR3_0_MA[7]		VSS	DDR3_0_MA[1]			DDR3_0_MON1P
7	VCCCPUVIDSI0_1P03							DDR3_0_DQSB[1]	DDR3_0_DQS[1]							VCCDDR_0_1P5	DDR3_0_MA[12]							VCCDDR_0_1P5	DDR3_0_MA[5]		
6																							VCCDDR_0_1P5				DDR3_0_MON1N
5		VCCCPUVIDSI0_1P03	VSS	VSS		VSS	VSS		VSS		VSS	VSS	VSS		VSS	VSS		VSS		DDR3_0_CKE[2]	DDR3_0_CKE[1]		DDR3_0_MON2P		VSS		VSS
4	VCCCPUVIDSI0_1P03	VCCCPUVIDSI0_1P03		DDR3_0_DQ[29]		DDR3_0_DQ[24]		DDR3_0_DQ[31]		VSS	VSS		DDR3_0_DQECC[0]		DDR3_0_DQSBECC[0]		DDR3_0_DQECC[3]		VSS	DDR3_0_CKE[0]		VCCDDR_0_1P5					
3	VCCCPUVIDSI0_1P03		VSS				DDR3_0_DQS[3]			DDR3_0_DQ[27]		DDR3_0_DQECC[4]				DDR3_0_DQECC[7]			VSS		DDR3_0_CKE[3]		DDR3_0_MON2N		VSS		VSS
2		VCCCPUVIDSI0_1P03		DDR3_0_DQ[28]		DDR3_0_DQ[25]		DDR3_0_DQ[30]			VSS		DDR3_0_DQECC[1]		DDR3_0_DQSECC[0]		DDR3_0_DQECC[2]										
1	VCCCPUVIDSI0_1P03			VSS			DDR3_0_DQSB[3]			DDR3_0_DQ[26]			DDR3_0_DQECC[5]			DDR3_0_DQECC[6]			VSS	VSS		VSS	VSS		VSS		VSS

§ §

36 Mechanical Characteristics

The SoC is manufactured as a 34 mm x 28 mm Flip-Chip Ball Grid Array (FCBGA) package and consists of a silicon die mounted face down on an organic substrate populated with 1283 solder balls on the bottom side. Capacitors are placed on the package top side in the area surrounding the die. Because die-side capacitors are electrically conductive, and only slightly shorter than the die height, care needs to be taken to avoid contacting the capacitors with electrically conductive materials. Doing so may short the capacitors and possibly damage the device or render it inactive.

The use of an insulating material between the capacitors and any thermal solution needs to be considered to prevent capacitor shorting. An exclusion, or keep out zone, surrounds the die and capacitors, and identifies the contact area for the package. Care needs to be taken to avoid contact with the package inside this area.

While package drawings are shown in this chapter, refer to the *Intel® Atom™ Processor C2000 Product Family for Microserver Thermal and Mechanical Specifications and Design Guide (TMSDG)* for details on package mechanical dimensions and tolerance and other key package attributes. The drawings shown here are for informational purposes and not meant to be the control documents for mechanical details of the package.

The following are the dimensions:

- Package parameters: 34 mm x 28 mm
- Ball Count: 1283

Figure 36-1. Topside Showing Capacitors and Marking Areas

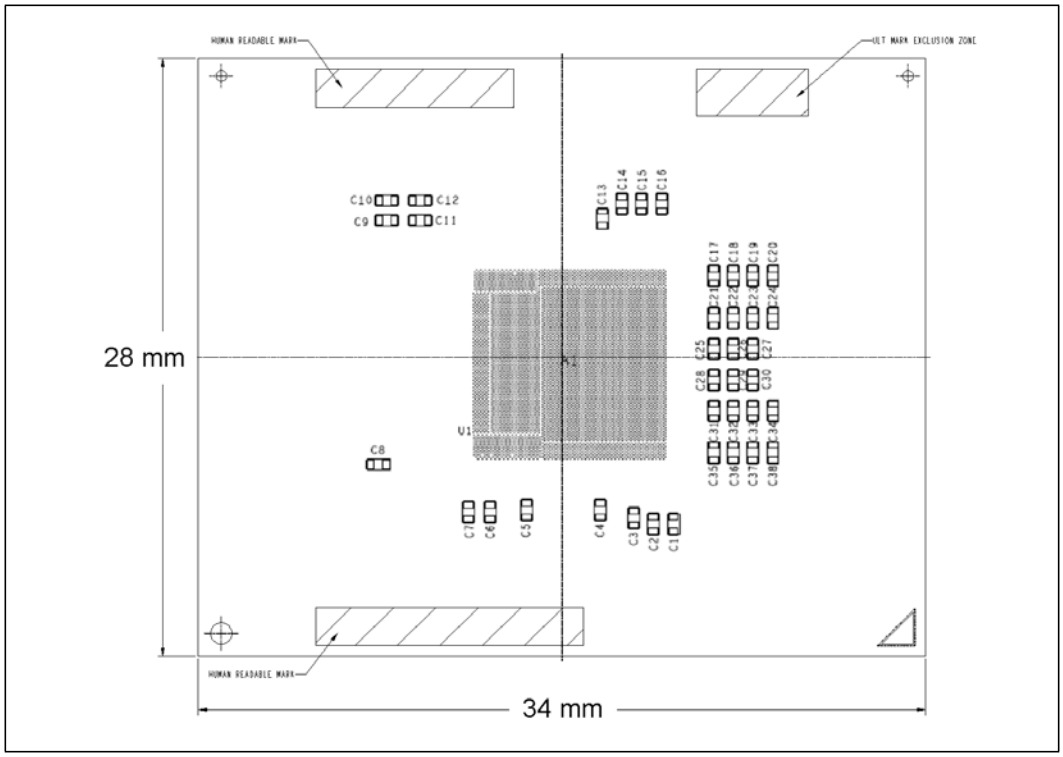
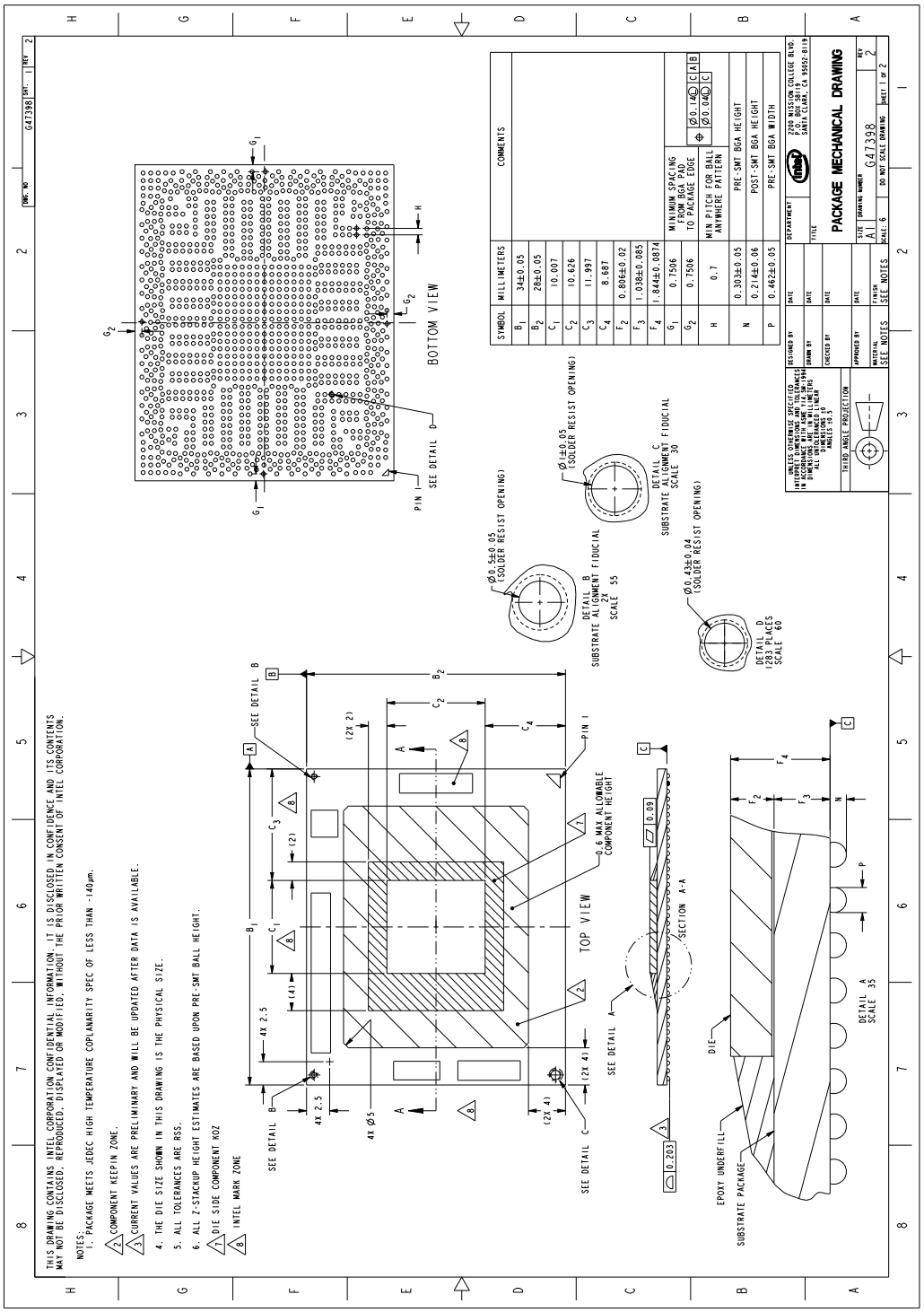


Figure 36-2. Package Mechanical Drawing



§ §

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Intel:

FH8065401488906S R1CS FH8065401488914S R1CV FH8065401488915S R1CU FH8065401488912S R1CT FH8065401488919S R1CR FH8065501516762S R1S9 FH8065501516709S R1CZ FH8065501516702S R1CW FH8065501516761S R1S8 FH8065501516708S R1CY FH8065501516710S R1D1 FH8065501516754S R1UN FH8065501516753S R1UM FH8065501516763S R1SA FH8065501516711S R1D2 FH8065501516768S R1VV

OCEAN CHIPS
Океан Электроники
Поставка электронных компонентов

Компания «Океан Электроники» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Поставка оригинальных импортных электронных компонентов напрямую с производств Америки, Европы и Азии, а так же с крупнейших складов мира;
- Широкая линейка поставок активных и пассивных импортных электронных компонентов (более 30 млн. наименований);
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Помощь Конструкторского Отдела и консультации квалифицированных инженеров;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Поставка электронных компонентов под контролем ВП;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- При необходимости вся продукция военного и аэрокосмического назначения проходит испытания и сертификацию в лаборатории (по согласованию с заказчиком);
- Поставка специализированных компонентов военного и аэрокосмического уровня качества (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Actel, Aeroflex, Peregrine, VPT, Syfer, Eurofarad, Texas Instruments, MS Kennedy, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits, General Dynamics и др.);

Компания «Океан Электроники» является официальным дистрибьютором и эксклюзивным представителем в России одного из крупнейших производителей разъемов военного и аэрокосмического назначения «**JONHON**», а так же официальным дистрибьютором и эксклюзивным представителем в России производителя высокотехнологичных и надежных решений для передачи СВЧ сигналов «**FORSTAR**».

AVIC JONHON

«JONHON» (основан в 1970 г.)

Разъемы специального, военного и аэрокосмического назначения:

(Применяются в военной, авиационной, аэрокосмической, морской, железнодорожной, горно- и нефтедобывающей отраслях промышленности)

«FORSTAR» (основан в 1998 г.)

ВЧ соединители, коаксиальные кабели, кабельные сборки и микроволновые компоненты:

(Применяются в телекоммуникациях гражданского и специального назначения, в средствах связи, РЛС, а так же военной, авиационной и аэрокосмической отраслях промышленности).

FORSTAR®

Телефон: 8 (812) 309-75-97 (многоканальный)
Факс: 8 (812) 320-03-32
Электронная почта: ocean@oceanchips.ru
Web: http://oceanchips.ru/
Адрес: 198099, г. Санкт-Петербург, ул. Калинина, д. 2, корп. 4, лит. А