Intel® 495 Series Chipset Family On-Package Platform Controller Hub (PCH)

Datasheet – Volume 1 of 2

October 2020

Revision 004

Document Number: 341080-004

Contents

Figures

Tables

Revision History

Revision Number	Description	Revision Date
001	Initial Release	August 2019
002	**Chapter 10, Electrical and Thermal Characteristics** • Updated Table 10-33	February 2020
003	**Chapter 22, PCI Express* (PCIe*)** • Added a note in "Precision Time Measurement (PTM)" section **Chapter 23, Power Management** • Updated description for SLP_A#, SLP_WLAN# and SLP_SUS#	July 2020
004	**Chapter 8, Power and Ground Signals** • Added note for VCC_VNNEXT_P1P05 **Chapter 23, Power Management** • Added note for SX_EXIT_HOLDOFF#	October 2020

§ §

1 Introduction

This document is intended for Original Equipment Manufacturers (OEMs), Original Design Manufacturers (ODM) and BIOS vendors creating products based on the Intel® 495 Series Chipset Family On-Package Platform Controller Hub (PCH).

Throughout this document, the Platform Controller Hub (PCH) is used as a general term and refers to all I/O SKUs, unless specifically noted otherwise.

This manual assumes a working knowledge of the vocabulary and principles of interfaces and architectures such as PCI Express* (PCIe), Universal Serial Bus (USB), Advance Host Controller Interface (AHCI), eXtensible Host Controller Interface (xHCI), and so on.

This manual abbreviates buses as B*n*, devices as D*n* and functions as F*n*. For example Device 31 Function 0 is abbreviated as D31:F0, Bus 1 Device 8 Function 0 is abbreviated as B1:D8:F0. Generally, the bus number will not be used, and can be considered to be Bus 0.

1.1 References

Specification	Document Number
Intel® 495 Series Chipset Family On-Package Platform Controller Hub (PCH) Datasheet, Volume 2 of 2	341081

1.2 Overview

The PCH provides extensive I/O support. Functions and capabilities include:

- ACPI Power Management Logic Support, Revision 5.0a
- PCI Express* Base Specification Revision 3.0
- Integrated Serial ATA Host controller 3.2, supports data transfer rates of up to 6Gb/s on all ports
- USB 3.2 Gen 2x1 (10 Gb/s) eXtensible Host Controller (xHCI)
- USB 3.2 Gen 1x1 (5 Gb/s) Dual Role (eXtensible Device Controller - xDCI) Capability
- embedded MultiMedia Card (eMMC*) Revision 5.1 Controller
- Serial Peripheral Interface (SPI)
- Enhanced Serial Peripheral Interface (eSPI)
- Flexible I/O—Allows some high speed I/O signals to be configured as PCIe*, SATA or USB 3.2
- General Purpose Input Output (GPIO)
- Interrupt controller
- Timer functions
- System Management Bus (SMBus) Specification, Version 2.0
- Integrated Clock Controller (ICC)/Real Time Clock Controller (RTCC)

- Intel® High Definition Audio and Intel® Smart Sound Technology (Intel® SST), supporting I^2C, Soundwire* and DMIC.
- Intel® Serial I/O UART Host controllers
- Intel® Serial I/O I^2C Host controllers
- Integrated 10/100/1000 Gigabit Ethernet MAC
- Integrated Sensor Hub (ISH)
- Supports Intel® Rapid Storage Technology (Intel® RST)
- Supports Intel® Active Management Technology (Intel® AMT)
- Supports Intel® Virtualization Technology for Directed I/O (Intel® VT-d)
- Supports Intel® Trusted Execution Technology (Intel® TXT)
- JTAG Boundary Scan support
- Intel® Trace Hub (Intel® TH) and Direct Connect Interface (DCI) for debug
- Supports Intel® CSME
- Supports Integrated connectivity (CNVi)

Note: Not all functions and capabilities may be available on all SKUs. The following table provides an overview of the PCH I/O capabilities.

1.3 PCH SKUs

Table 1-1. PCH I/O Capabilities

Interface	PCH-LP
CPU Interface	OPI x8, up to 4 GT/s
eSPI	1 CS# with in-band ALERT#
I^2C	6
UART	3
Generic SPI(GSPI)	3
Integrated Sensor Hub (ISH 5.2)	3 I^2C, 2 UART, 1 GSPI

Table 1-2. PCH SKUs

Features	SKU	
	Premium U	Premium Y
SATA Ports(all 6 Gb/s capable)	Up to 3	Up to 2
PCIe* Lanes	Up to 16 Gen3 lanes (6 devices maximum)	Up to 14 Gen3 lanes (5 devices maximum)
USB 2.0 Ports	10 HS(USB 2.0)	6 HS(USB 2.0)
USB 3.2 Ports	Up to 6 SuperSpeed USB 10 Gbps (USB 3.2 Gen 1x1/Gen 2x1)	Up to 6 SuperSpeed USB 10 Gbps (USB 3.2 Gen 1x1/Gen 2x1)
SDXC	SDXC 3.0	SDXC 3.0

Table 1-3. PCH HSIO Details

Flex I/O Lane	SKU	
	Premium U	Premium Y
0	USB 3.2 Gen 1x1/2x1 #1, PCIe* #1	USB 3.2 Gen 1x1/2x1 #1, PCIe #1
1	USB 3.2 Gen 1x1/2x1 #2, PCIe #2	USB 3.2 Gen 1x1/2x1 #2, PCIe #2
2	USB 3.2 Gen 1x1/2x1 #3, PCIe #3	USB 3.2 Gen 1x1/2x1 #3, PCIe #3
3	USB 3.2 Gen 1x1/2x1 #4, PCIe #4	USB 3.2 Gen 1x1/2x1 #4, PCIe #4
4	USB 3.2 Gen 1x1/2x1 #5, PCIe #5	USB 3.2 Gen 1x1/2x1 #5, PCIe #5
5	USB 3.2 Gen 1x1/2x1 #6, PCIe #6	USB 3.2 Gen 1x1/2x1 #6, PCIe #6
6	PCIe #7, GbE	PCIe #7, GbE
7	PCIe #8, GbE	PCIe #8, GbE
8	PCIe #9, GbE	PCIe #9, GbE
9	PCIe #10	PCIe #10
10	PCIe #11, SATA #0	PCIe #11, SATA #0
11	PCIe #12, SATA #1A	PCIe #12, SATA #1A
12	PCIe #13, GbE	PCIe #13, GbE
13	PCIe #14, GbE	PCIe #14, GbE
14	PCIe #15, SATA #1B	NA
15	PCIe #16, SATA #2	NA

§ §

2 PCH Controller Device IDs

2.1 Device and Revision ID Table

The Revision ID (RID) register is an 8-bit register located at offset 08h in the PCI header of every PCI/PCIe* function. The RID register is used by software to identify a particular component stepping when a driver change or patch unique to that stepping is needed. The RID register reports one of the two possible values: Stepping Revision Identification (SRID) or Compatible Revision ID (CRID). The default power-on value for the RID register is SRID. The assigned value is based on the product's stepping. CRID is intended for the corporate Intel® Stable Image Platform Program (Intel® SIPP). CRID is normally identical to the SRID value of a previous production stepping of the product with which the new stepping is deemed "compatible". Intel® SIPP allows an OS image built on the earlier stepping to be used on any new "compatible" stepping(s). Three CRID values are possible and can be used to manage software images.

Note: SRID and CRID are not addressable PCI registers. The SRID and CRID value are reflected through the RID register when appropriately selected.

Following reset, the SRID value can be read from the RID registers of all PCH devices and functions. To select either SRID or CRID to be reflected in the RID registers, BIOS needs to write appropriate value into the Configured Revision ID (CRID) register located in the PMC MMIO space. Refer to Vol2 of this document for definition details of the register. BIOS must write this register with the appropriate value after PLTRST# events.

After CRID is selected and applied by BIOS, software will not be able to obtain the original SRID value of the PCH by reading the PCH RID registers. Customers implementing CRID whom also want to determine the SRID in runtime may develop their own tool. For example, BIOS can capture the SRID value before BIOS applies CRID and store that value in a runtime accessible place (that is, SMBIOS, ACPI Type 4 Memory, NVRAM, CMOS) so that it can be read by the customer tool later. Alternatively, the BIOS can store the SRID value and display this information in BIOS setup while reporting that CRID is enabled.

BIOS needs to check CRID_UIP bit (in PMC MMIO space) as a part of the update flow. PMC HW sets this bit to indicate that SetID broadcast flow has been requested by BIOS. This bit is cleared by PMC FW only when the completion/s of SetIDVal message is received by PMC. BIOS is required to read this bit as cleared before writing to the CRID register (to request a CRID update). BIOS is also required to poll on reads to this bit until it sees the bit as cleared after BIOS has written to the CRID register.

Table 2-1. PCH-U/Y Device and Revision ID Table (Sheet 1 of 2)

Dev ID	Device Function - Device Description	A0 SRID	Note
3480 - 349F	D31:F0 - eSPI Controller		**PCH Device IDs:** PCH-LP U Full featured engineering sample: 3481 PCH-LP U Premium: 3482 PCH-LP Y Full featured engineering sample: 3486 PCH-LP Y Premium: 3487
34A0	D31:F1 - P2SB		
34A1	D31:F2 - PMC		
34A3	D31:F4 - SMBus		
34A4	D31:F5 - SPI (flash) Controller		
15E1	D31:F6 - GbE Controller		GbE Controller: Corporate/ Intel® vPro™ (Default)
15E2	D31:F6 - GbE Controller		GbE Controller: Consumer
15E3	D31:F7 - Intel® Trace Hub (Intel® TH)		
34A8	D30:F0 - UART #0		
34A9	D30:F1 - UART #1		
34AA	D30:F2 - GSPI #0		
34AB	D30:F3 - GSPI #1		
34B0	D29:F0 - PCI Express* Root Port #9		
34B1	D29:F1 - PCI Express Root Port #10		
34B2	D29:F2 - PCI Express Root Port #11		
34B3	D29:F3 - PCI Express Root Port #12		
34B4	D29:F4 - PCI Express Root Port #13		
34B5	D29:F5 - PCI Express Root Port #14		
34B6	D29:F6 - PCI Express Root Port #15		
34B7	D29:F7 - PCI Express Root Port #16		
34B8	D28:F0 - PCI Express Root Port #1		
34B9	D28:F1 - PCI Express Root Port #2		
34BA	D28:F2 - PCI Express Root Port #3		
34BB	D28:F3 - PCI Express Root Port #4		
34BC	D28:F4 - PCI Express Root Port #5		
34BD	D28:F5 - PCI Express Root Port #6		
34BE	D28:F6 - PCI Express Root Port #7		
34BF	D28:F7 - PCI Express Root Port #8		
34C4	D26:F0 - eMMC*		
34C5	D25:F0 - I^2C Controller #4		
34C6	D25:F1 - I^2C Controller #5		
34C7	D25:F2 - UART #2		
34C8 - 34CF	D31:F3 - AVS (Audio, Voice, Speech)		

Table 2-1. PCH-U/Y Device and Revision ID Table (Sheet 2 of 2)

Dev ID	Device Function - Device Description	A0 SRID	Note
34D3	D23:F0 - SATA Controller (AHCI)		
34D5	D23:F0 - SATA Controller (RAID 0/1/5/10) - NOT premium		
34D7	D23:F0 - SATA Controller (RAID 0/1/5/10) - premium		
282A	D23:F0 - SATA Controller (RAID 0/1/5/10) - In-box Compatible ID		
34E0	D22:F0 - Intel® CSME: HECI #1		
34E1	D22:F1 - Intel® CSME: HECI #2		
34E2	D22:F2 - Intel® CSME: IDE Redirection (IDER-R)		
34E3	D22:F3 - Intel® CSME: Keyboard and Text (KT) Redirection		
34E4	D22:F4 - Intel® CSME: HECI #3		
34E5	D22:F5 - Intel® CSME: HECI #4		
34E8	D21:F0 - I^2C Controller #0		
34E9	D21:F1 - I^2C Controller #1		
34EA	D21:F2 - I^2C Controller #2		
34EB	D21:F3 - I^2C Controller #3		
34ED	D20:F0 - USB 3.2 Gen 2x1 (10 Gb/s) xHCI Host Controller		
34EE	D20:F1 - USB 3.2 Gen 1x1 (5 Gb/s) Device Controller (xDCI)		
34EF	D20:F2 - Shared SRAM		
34F0 - 34F3	D20:F3 - CNVi		
34F8	D20:F5 - SDXC		
34FB	D18:F6 - GSPI #2		
34FC	D18:F0 - Integrated Sensor Hub		

§ §

3 Flexible I/O

3.1 Acronyms

Acronyms	Description
USB	Universal Serial Bus
PCIe*	PCI Express* (Peripheral Component Interconnect Express*)
GbE	Gigabit Ethernet
SATA	Serial Advanced Technology Attachment

3.2 PCH U SKU

Figure 3-1. High Speed I/O (HSIO) Lane Multiplexing in PCH U

Flex I/O Lane	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
High Speed I/O (HSIO) Type and Lane	USB 3.2 Gen 1x1/2x1 #1	USB 3.2 Gen 1x1/2x1 #2	USB 3.2 Gen 1x1/2x1 #3	USB 3.2 Gen 1x1/2x1 #4	USB 3.2 Gen 1x1/2x1 #5	USB 3.2 Gen 1x1/2x1 #6	PCIe* #7	PCIe* #8	PCIe* #9	PCIe* #10	PCIe* #11	PCIe* #12	PCIe* #13	PCIe* #14	PCIe* #15	PCIe* #16
	PCIe* #1	PCIe* #2	PCIe* #3	PCIe* #4	PCIe* #5	PCIe* #6	GbE	GbE	GbE		SATA 0	SATA 1a	GbE	GbE	SATA 1b	SATA 2
PCIe* Remap Support	No				No				Yes				Yes			

Note: Some lane multiplexing capabilities are not available on all SKUs. Refer to the PCH SKU section for specific SKU details.

The 16 HSIO lanes on U PCH-LP supports the following configurations:

1. Up to 16 PCIe* Lanes
   — A maximum of six PCIe* Root Ports (or devices) can be enabled
   - When a GbE Port is enabled, the maximum number of PCIe* Root Ports (or devices) that can be enabled reduces based off the following:
     --> Max PCIe* Root Ports (or devices) = 6 - GbE (0 or 1)

— PCIe* Lanes 1-4 (PCIe* Controller #1), 5-8 (PCIe* Controller #2), 9-12 (PCIe* Controller #3), and 13-16 (PCIe* Controller #4) can be individually configured

2. Up to four SATA Lanes
   — A maximum of three SATA Ports (or devices) can be enabled
   — SATA Lane 1 has the flexibility to be mapped to Flex I/O Lane 11 or 14
3. Up to six USB 3.2 Gen 1x1/2x1 Lanes
   — A maximum of six USB 3.2 Gen 1x1/2x1 Ports (or devices) can be enabled
   — USB 3.2 Gen 1x1 = 5 GT/s
   — USB 3.2 Gen 2x1 = 10 GT/s
4. Up to five GbE Lanes
   — A maximum of one GbE Port (or device) can be enabled
5. Supports up to two PCIe* Remapped Storage Devices
   — Refer the Chapter 22, "PCI Express* (PCIe*)" for the PCH PCIe* Controllers, configurations, and lanes that can be used for PCIe* Remapped Storage Device support
6. For unused SATA/PCIe* and USB 3.1/PCIe* Combo Lanes, the unused lanes must be statically assigned to PCIe*, SATA, or USB 3.1 via the SATA/PCIe* and USB 3.1/ PCIe* Combo Port Soft Straps .

3.3 PCH Y SKU

Figure 3-2. High Speed I/O (HSIO) Lane Multiplexing in PCH Y

Flex I/O Lane	0	1	2	3	4	5	6	7	8	9	10	11	12	13
High Speed I/O (HSIO) Type and Lane	USB 3.2 Gen 1x1/2x1 #1	USB 3.2 Gen 1x1/2x1 #2	USB 3.2 Gen 1x1/2x1 #3	USB 3.2 Gen 1x1/2x1 #4	USB 3.2 Gen 1x1/2x1 #5	USB 3.2 Gen 1x1/2x1 #6	PCIe* #7	PCIe* #8	PCIe* #9	PCIe* #10	PCIe* #11	PCIe* #12	PCIe* #13	PCIe* #14
	PCIe* #1	PCIe* #2	PCIe* #3	PCIe* #4	PCIe* #5	PCIe* #6	GbE	GbE	GbE		SATA 0	SATA 1a	GbE	GbE
PCIe* Remap Support	No				No				Yes				Yes	

The 14 HSIO lanes on Y PCH-LP supports the following configurations:

1. Up to 14 PCIe* Lanes
   — A maximum of 5 PCIe* Root Ports (or devices) can be enabled
      - When a GbE Port is enabled, the maximum number of PCIe* Root Ports (or devices) that can be enabled reduces based off the following:
        --> Max PCIe* Root Ports (or devices) = 5 - GbE (0 or 1)

   — PCIe* Lanes 1-4 (PCIe* Controller #1), 5-8 (PCIe* Controller #2), 9-12 (PCIe* Controller #3), and 13-14 (PCIe* Controller #4) can be individually configured
2. Up to two SATA Lanes
   — A maximum of two SATA Ports (or devices) can be enabled
3. Up to six USB 3.2 Gen 1x1/2x1 Lanes
   — A maximum of six USB 3.2 Gen 1x1/2x1 Ports (or devices) can be enabled
   — USB 3.2 Gen 1x1 = 5 GT/s
   — USB 3.2 Gen 2x1 = 10 GT/s
4. Up to five GbE Lanes
   — A maximum of one GbE Port (or device) can be enabled
5. Supports up to two PCIe* Remapped Storage Devices
   — Refer the Chapter 22, "PCI Express* (PCIe*)" for the PCH PCIe* Controllers, configurations, and lanes that can be used for PCIe* Remapped Storage Device support
6. For unused SATA/PCIe* and USB 3.1/PCIe* Combo Lanes, the unused lanes must be statically assigned to PCIe*, SATA, or USB 3.1 via the SATA/PCIe* and USB 3.1/PCIe* Combo Port Soft Straps.

3.4 Overview/Functional Description

Flexible Input/Output (I/O) is a technology that allows some of the PCH High Speed I/O (HSIO) lanes to be configured for connection to a Gigabit Ethernet (GbE) Controller, a PCIe* Controller, a Extensible Host Controller Interface (XHCI) USB 3.1 Controller, or a Advanced Host Controller Interface (AHCI) SATA Controller. Flexible I/O enables customers to optimize the allocation of the PCH HSIO interfaces to better meet the I/O needs of their system.

Note: Some Flexible I/O multiplexing capabilities are not available on all SKUs. Refer to the Section 1.3, "PCH SKUs" in the "Introduction" chapter for specific SKU implementation details.

3.4.1 Flexible I/O Lane Selection

HSIO lane configuration and type is statically selected by soft straps, which are managed through the Platform Flash Image Tool, available as part of Intel® CSME releases. Refer to the SPI Programming Guide documentation for details on how to configure the Flexible I/O lanes via soft straps.

Note: It is the responsibility of the platform designers to configure the lane muxing and soft straps correctly without any conflict. The hardware behavior is undefined if this scenario ever happens.

3.4.1.1 PCIe*/SATA Lane Selection

In addition to static configuration via soft straps, Flexible I/O Lanes that have PCIe*/SATA multiplexing can be configured via SATAXPCIE signaling to support implementation like SATA Express or mSATA, where the port configuration is selected by the type of the add-in card that is used. Refer to the Platform SPI Programming Guide for more details on how to configure SATAXPCIE for SATA/PCIe* lane selection.

§ §

4 Memory Mapping

This section describes (from the processor perspective) the memory ranges that the PCH decodes.

4.1 Functional Description

4.1.1 PCI Devices and Functions

The PCH incorporates a variety of PCI devices and functions, as shown in the following table. If for some reason, the particular system platform does not want to support any one of the Device Functions, with the exception of D30:F0, they can individually be disabled. The integrated Gigabit Ethernet controller will be disabled if no Platform LAN Connect component is detected (Refer Chapter 19, "Gigabit Ethernet Controller"). When a function is disabled, it does not appear at all to the software. A disabled function will not respond to any register reads or writes, insuring that these devices appear hidden to software.

Note: The reference to DMI for LP SKUs is On Package DMI (OPI).

Table 4-1. PCI Devices and Functions (Sheet 1 of 2)

Devices and Function	Description
Bus M: Device 31: Function 0	eSPI Interface
Bus M: Device 31: Function 1	P2SB
Bus M: Device 31: Function 2	PMC
Bus M: Device 31: Function 3	AVS (Audio, Voice, Speech)
Bus M: Device 31: Function 4	SMBus Controller
Bus M: Device 31: Function 5	SPI
Bus M: Device 31: Function 6	GbE Controller
Bus M: Device 31: Function 7	Intel® Trace Hub (Intel® TH)
Bus M: Device 30: Function 0	UART #0
Bus M: Device 30: Function 1	UART #1
Bus M: Device 30: Function 2	GSPI #0
Bus M: Device 30: Function 3	GSPI #1
Bus M: Device 29: Function 0	PCI Express* Port 9
Bus M: Device 29: Function 1	PCI Express Port 10
Bus M: Device 29: Function 2	PCI Express Port 11
Bus M: Device 29: Function 3	PCI Express Port 12
Bus M: Device 29: Function 4	PCI Express Port 13
Bus M: Device 29: Function 5	PCI Express Port 14
Bus M: Device 29: Function 6	PCI Express Port 15
Bus M: Device 29: Function 7	PCI Express Port 16
Bus M: Device 28: Function 0	PCI Express Port 1

Table 4-1. PCI Devices and Functions (Sheet 2 of 2)

Devices and Function	Description
Bus M: Device 28: Function 1	PCI Express Port 2
Bus M: Device 28: Function 2	PCI Express Port 3
Bus M: Device 28: Function 3	PCI Express Port 4
Bus M: Device 28: Function 4	PCI Express Port 5
Bus M: Device 28: Function 5	PCI Express Port 6
Bus M: Device 28: Function 6	PCI Express Port 7
Bus M: Device 28: Function 7	PCI Express Port 8
Bus M: Device 26: Function 0	eMMC*
Bus M: Device 25: Function 0	I^2C Controller #4
Bus M: Device 25: Function 1	I^2C Controller #5
Bus M: Device 25: Function 2	UART Controller #2
Bus M: Device 23: Function 0	SATA Controller
Bus M: Device 22: Function 0	Intel® CSME: HECI #1
Bus M: Device 22: Function 1	Intel® CSME: HECI #2
Bus M: Device 22: Function 2	Intel® CSME: IDE-Redirection (IDE-R)
Bus M: Device 22: Function 3	Intel® CSME: Keyboard and Text (KT) Redirection
Bus M: Device 22: Function 4	Intel® CSME: HECI #3
Bus M: Device 22: Function 5	Intel® CSME: HECI #4
Bus M: Device 22: Function 7	Intel® CSME: WLAN
Bus M: Device 21: Function 0	I^2C Controller #0
Bus M: Device 21: Function 1	I^2C Controller #1
Bus M: Device 21: Function 2	I^2C Controller #2
Bus M: Device 21: Function 3	I^2C Controller #3
Bus M: Device 21: Function 6	vTIO Phantom Function for xHCI
Bus M: Device 21: Function 7	Intel® Trace Hub Phantom (ACPI) Function
Bus M: Device 20: Function 0	xHCI - USB Host Controller
Bus M: Device 20: Function 1	xDCI - USB Device Controller
Bus M: Device 20: Function 2	Shared SRAM
Bus M: Device 20: Function 3	CNVi: Wi-Fi*
Bus M: Device 20: Function 4	vTIO Phantom (ACPI) Function
Bus M: Device 20: Function 5	SCS3: SDXC
Bus M: Device 18: Function 0	Integrated Sensor Hub
Bus M: Device 18: Function 2	Intel® CSME: PMT Phantom (ACPI) function
Bus M: Device 18: Function 4	Intel® CSME: fTPM DMA Phantom (ACPI) Function
Bus M: Device 18: Function 6	GSPI #2

4.1.2 Fixed I/O Address Ranges

Below table shows the Fixed I/O decode ranges from the processor perspective.

Note: For each I/O range, there may be separate behavior for reads and writes.

DMI cycles that go to target ranges that are marked as Reserved will be handled by the PCH; writes are ignored and reads will return all 1s. The P2SB will claim many of the fixed I/O accesses and forward those transactions over IOSF-SB to their functional target.

Address ranges that are not listed or marked Reserved are NOT positively decoded by the PCH (unless assigned to one of the variable ranges) and will be internally terminated by the PCH.

Table 4-2. Fixed I/O Ranges Decoded by PCH (Sheet 1 of 3)

I/O Address	Read Target	Write Target	Internal Unit (unless[E]: External)[2]	Separate Enable/Disable
20h – 21h	Interrupt Controller	Interrupt Controller	Interrupt	None
24h – 25h	Interrupt Controller	Interrupt Controller	Interrupt	None
28h – 29h	Interrupt Controller	Interrupt Controller	Interrupt	None
2Ch – 2Dh	Interrupt Controller	Interrupt Controller	Interrupt	None
2E-2F	Super I/O	Super I/O	[E] Forwarded to eSPI	Yes. IOE.SE
30h – 31h	Interrupt Controller	Interrupt Controller	Interrupt	None
34h – 35h	Interrupt Controller	Interrupt Controller	Interrupt	None
38h – 39h	Interrupt Controller	Interrupt Controller	Interrupt	None
3Ch – 3Dh	Interrupt Controller	Interrupt Controller	Interrupt	None
40h	Timer/Counter	Timer/Counter	8254 Timer	None
42h-43h	Timer/Counter	Timer/Counter	8254 Timer	None
4E-4F	Microcontroller	Microcontroller	[E] Forwarded to eSPI	Yes. IOE.ME2
50h	Timer/Counter	Timer/Counter	8254 Timer	None
52h-53h	Timer/Counter	Timer/Counter	8254 Timer	None
60h	Keyboard Controller	Keyboard Controller	[E] Forwarded to eSPI	Yes, with 64h. IOE.KE
61h	NMI Controller	NMI Controller	CPU I/F	None
62h	Microcontroller	Microcontroller	[E] Forwarded to eSPI	Yes, with 66h. IOE.ME1
63h	NMI Controller[1]	NMI Controller[1]	CPU I/F	Yes, alias to 61h. GIC.P61AE
64h	Keyboard Controller	Keyboard Controller	[E] Forwarded to eSPI	Yes, with 60h. IOE.KE
65h	NMI Controller[1]	NMI Controller[1]	CPU I/F	Yes, alias to 61h. GIC.P61AE
66h	Microcontroller	Microcontroller	[E] Forwarded to eSPI	Yes, with 62h. IOE.ME1

Table 4-2. Fixed I/O Ranges Decoded by PCH (Sheet 2 of 3)

I/O Address	Read Target	Write Target	Internal Unit (unless[E]: External)[2]	Separate Enable/Disable
67h	NMI Controller[1]	NMI Controller[1]	CPU I/F	Yes, alias to 61h. GIC.P61AE
70h	RTC Controller	NMI and RTC Controller	RTC	None
71h	RTC Controller	RTC Controller	RTC	None
72h	RTC Controller	RTC Controller	RTC	None. Alias to 70h if RC.UE[4]=0, else 72h
73h	RTC Controller	RTC Controller	RTC	None. Alias to 71h if RC.UE='0', else 73h
74h	RTC Controller	RTC Controller	RTC	None
75h	RTC Controller	RTC Controller	RTC	None
76h-77h	RTC Controller	RTC Controller	RTC	None. Alias to 70h-71h if RC.UE=0, else 76h-77h
80h[3]	eSPI or PCIe*	eSPI or PCIe	Read: [E] eSPI or PCIe Write: [E] eSPI or [E] PCIe	None. PCIe if GCS.RPR='1', else eSPI
84h - 86h	eSPI or PCIe	eSPI or PCIe	Read: [E] eSPI or PCIe Write: [E] eSPI or [E] PCIe	None. PCIe if GCS.RPR='1', else eSPI
88h	eSPI or PCIe	eSPI or PCIe	Read: [E] eSPI or PCIe Write: [E] eSPI or [E] PCIe	None. PCIe if GCS.RPR='1', else eSPI
8Ch - 8Eh	eSPI or PCIe	eSPI or PCIe	Read: [E] eSPI or PCIe Write: [E] eSPI or [E] PCIe	None. PCIe if GCS.RPR='1', else eSPI
90h	eSPI	eSPI	Read: [E] eSPI Write: [E] eSPI	None. Alias to 80h
92h	Reset Generator	Reset Generator	CPU I/F	None
94h - 96h	eSPI	eSPI	Read: [E] eSPI Write: [E] eSPI	None. Alias to 8xh
98h	eSPI	eSPI	Read: [E] eSPI Write: [E] eSPI	None. Alias to 88h

Table 4-2. Fixed I/O Ranges Decoded by PCH (Sheet 3 of 3)

I/O Address	Read Target	Write Target	Internal Unit (unless[E]: External)[2]	Separate Enable/Disable
9Ch - 9Eh	eSPI	eSPI	Read: [E] eSPI Write: [E] eSPI	None. Alias to 8xh
A0h - A1h	Interrupt Controller	Interrupt Controller	Interrupt	None
A4h - A5h	Interrupt Controller	Interrupt Controller	Interrupt	None
A8h - A9h	Interrupt Controller	Interrupt Controller	Interrupt	None
ACh - ADh	Interrupt Controller	Interrupt Controller	Interrupt	None
B0h - B1h	Interrupt Controller	Interrupt Controller	Interrupt	None
B2h - B3h	Power Management	Power Management	Power Management	None
B4h - B5h	Interrupt Controller	Interrupt Controller	Interrupt	None
B8h - B9h	Interrupt Controller	Interrupt Controller	Interrupt	None
BCh - BDh	Interrupt Controller	Interrupt Controller	Interrupt	None
200-207h	Gameport Low	Gameport Low	Forwarded to eSPI	Yes. IOE.LGE
208-20Fh	Gameport High	Gameport High	Forwarded to eSPI	Yes. IOE.HGE
4D0h – 4D1h	Interrupt Controller	Interrupt Controller	Interrupt Controller	None
CF9h	Reset Generator	Reset Generator	Interrupt controller	None

Notes:
1. Only if the Port 61 Alias Enable bit (GIC.P61AE) bit is set. Otherwise, the cycle is internally terminated by the PCH.
2. Destination of eSPI when eSPI Disabled pin strap is 0.
3. This includes byte, word or double-word (DW) access at I/O address 80h.

4.1.3 Variable I/O Decode Ranges

Table 4-3, "Variable I/O Decode Ranges" shows the Variable I/O Decode Ranges. They are set using Base Address Registers (BARs) or other configuration bits in the various configuration spaces. The PnP software (PCI or ACPI) can use their configuration mechanisms to set and adjust these values.

Warning: The Variable I/O Ranges should not be set to conflict with the Fixed I/O Ranges. There may some unpredictable results if the configuration software allows conflicts to occur. The PCH does not perform any checks for conflicts.

Table 4-3. Variable I/O Decode Ranges (Sheet 1 of 2)

Range Name[1]	Mappable	Size (Bytes)	Target
ACPI	Anywhere in 64K I/O Space	256	Power Management
IDE Bus Master	Anywhere in 64K I/O Space	16 or 32 Bytes	Intel® AMT IDE-R
SMBus	Anywhere in 64K I/O Space	32	SMB Unit
TCO	Anywhere in 64K I/O Space	32	SMB Unit
Parallel Port	3 ranges in 64K I/O Space	8	eSPI
Serial Port 1	8 Ranges in 64K I/O Space	8	eSPI
Serial Port 2	8 Ranges in 64K I/O Space	8	eSPI

Table 4-3. Variable I/O Decode Ranges (Sheet 2 of 2)

Range Name[1]	Mappable	Size (Bytes)	Target
Serial Port 3	8 Ranges in 64K I/O space	8	eSPI
Floppy Disk Controller	2 Ranges in 64K I/O Space	8	eSPI
IO Trapping Ranges	Anywhere in 64K I/O Space	1 to 256 Bytes	Trap
Serial ATA Index/Data Pair	Anywhere in 64K I/O Space	16	SATA Host Controller
PCI Express* Root Ports	Anywhere in 64K I/O Space	I/O Base/Limit	PCI Express Root Ports 1-16
Keyboard and Text (KT)	Anywhere in 64K I/O Space	8	Intel® AMT Keyboard and Text
Notes: 1. All ranges are decoded directly from OPI.			

4.2 Memory Map

Below table shows (from the Processor perspective) the memory ranges that the PCH will decode. Cycles that arrive from DMI that are not directed to any of the internal memory targets that decode directly from DMI will be master aborted.

PCIe* cycles generated by external PCIe masters will be positively decoded unless they fall in the PCI-PCI bridge memory forwarding ranges (those addresses are reserved for PCI peer-to-peer traffic). If the cycle is not in the internal LAN controller's range, it will be forwarded up to DMI. Software must not attempt locks to the PCH's memory-mapped I/O ranges.

Note: Total ports are different for the different SKUs.

Table 4-4. PCH Memory Decode Ranges (Processor Perspective) (Sheet 1 of 3)

Memory Range	Target	Dependency/Comments
000E 0000 - 000E FFFF	eSPI or SPI	Bit 6 in BIOS Decode Enable Register is set
000F 0000 - 000F FFFF	eSPI or SPI	Bit 7 in BIOS Decode Enable Register is set
FECX X000 - FECX X040	I/O(x)APIC inside PCH	XX controlled via APIC Range Select (ASEL) field and APIC Enable (AEN) bit.
FECX X000 - FECX XFFF	PCIe* port N (N=1 to 16)	X controlled via PCIe root port N IOxAPIC Range Base/Limit registers and Port N I/OxApic Enable (PAE) is set
FEC1 0000 - FEC1 7FFF	PCIe port 1	PCIe root port 1 I/OxApic Enable (PAE) is set
FEC1 8000 - FEC1 FFFF	PCIe port 2	PCIe root port 2 I/OxApic Enable (PAE) is set
FEC2 0000 - FEC2 7FFF	PCIe port 3	PCIe root port 3 I/OxApic Enable (PAE) is set
FEC2 8000 - FEC2 FFFF	PCIe port 4	PCIe root port 4 I/OxApic Enable (PAE) is set
FEC3 0000 - FEC3 7FFF	PCIe port 5	PCIe root port 5 I/OxApic Enable (PAE) is set
FEC3 8000 - FEC3 FFFF	PCIe port 6	PCIe root port 6 I/OxApic Enable (PAE) is set
FEC4 0000 - FEC4 7FFF	PCIe port 7	PCIe root port 7 I/OxApic Enable (PAE) is set
FEC4 8000 - FEC4 FFFF	PCIe port 8	PCIe root port 8 I/OxApic Enable (PAE) is set
FEC5 0000 - FEC5 7FFF	PCIe port 9	PCIe root port 9 I/OxApic Enable (PAE) is set
FEC5 8000 - FEC5 FFFF	PCIe port 10	PCIe root port 10 I/OxApic Enable (PAE) is set
FEC6 0000 - FEC6 7FFF	PCIe port 11	PCIe root port 11 I/OxApic Enable (PAE) is set
FEC6 8000 - FEC6 FFFF	PCIe port 12	PCIe root port 12 I/OxApic Enable (PAE) is set

Table 4-4. PCH Memory Decode Ranges (Processor Perspective) (Sheet 2 of 3)

Memory Range	Target	Dependency/Comments
FEC7 0000 - FEC7 7FFF	PCIe port 13	PCIe root port 13 I/OxApic Enable (PAE) is set
FEC7 8000 - FEC7 FFFF	PCIe port 14	PCIe root port 14 I/OxApic Enable (PAE) is set
FEC8 0000 - FEC8 7FFF	PCIe port 15	PCIe root port 15 I/OxApic Enable (PAE) is set
FEC8 8000 - FEC8 FFFF	PCIe port 16	PCIe root port 16 I/OxApic Enable (PAE) is set
FEC9 0000 - FEC9 7FFF	PCIe port 17	PCIe root port 17 I/OxApic Enable (PAE) is set
FEC9 8000 - FEC9 FFFF	PCIe port 18	PCIe root port 18 I/OxApic Enable (PAE) is set
FECA 0000 - FECA 7FFF	PCIe port 19	PCIe root port 19 I/OxApic Enable (PAE) is set
FECA 8000 - FECA FFFF	PCIe port 20	PCIe root port 20 I/OxApic Enable (PAE) is set
FEF0 0000 - FEFF FFFF	eSPI or SPI	uCode Patch Region Enable UCPR.UPRE is set
FFC0 0000 - FFC7 FFFF FF80 0000 - FF87 FFFF	eSPI or SPI	Bit 8 in BIOS Decode Enable Register is set
FFC8 0000 – FFCF FFFF FF88 0000 - FF8F FFFF	eSPI or SPI	Bit 9 in BIOS Decode Enable Register is set
FFD0 0000 - FFD7 FFFF FF90 0000 - FF97 FFFF	eSPI or SPI	Bit 10 in BIOS Decode Enable Register is set
FFD8 0000 – FFDF FFFF FF98 0000 - FF9F FFFF	eSPI or SPI	Bit 11 in BIOS Decode Enable Register is set
FFE0 0000 - FFE7 FFFF FFA0 0000 - FFA7 FFFF	eSPI or SPI	Bit 12 in BIOS Decode Enable Register is set
FFE8 0000 – FFEF FFFF FFA8 0000 – FFAF FFFF	eSPI or SPI	Bit 13 in BIOS Decode Enable Register is set
FFF0 0000 - FFF7 FFFF FFB0 0000 - FFB7 FFFF	eSPI or SPI	Bit 14 in BIOS Decode Enable Register is set
FFFC 0000 - FFFF FFFF	eSPI, SPI, or Intel® CSME	Always enabled. Refer to Table 4-5 - Boot Block Update Scheme for swappable ranges
FFF8 0000 - FFFB FFFF FFB8 0000 - FFBF FFFF	eSPI or SPI	Always enabled. Refer to Table 4-5 - Boot Block Update Scheme for swappable ranges
FF70 0000 - FF7F FFFF FF30 0000 - FF3F FFFF	eSPI or SPI	Bit 3 in BIOS Decode Enable Register is set
FF60 0000 - FF6F FFFF FF20 0000 - FF2F FFFF	eSPI or SPI	Bit 2 in BIOS Decode Enable Register is set
FF50 0000 - FF5F FFFF FF10 0000 - FF1F FFFF	eSPI or SPI	Bit 1 in BIOS Decode Enable Register is set
FF40 0000 - FF4F FFFF FF00 0000 - FF0F FFFF	eSPI or SPI	Bit 0 in BIOS Decode Enable Register is set
FED0 X000 - FED0 X3FF	HPET	BIOS determines “fixed” location which is one of four 1KB ranges where X (in the first column) is 0h, 1h, 2h, or 3h.
FED4 0000 - FED4 7FFF	SPI (set by strap)	TPM and Trusted Mobile KBC
FED4 C000 - FED4 FFFF	PCH Internal (PSF Error Handler)	Always enabled
FED6 0000 – FED6 1FFF	PCH Internal (Intel® Trace Hub (Intel® TH)/xHCI)	Always enabled
FED6 2000 – FED6 3FFF	xHCI (CPU)	Fixed range in CPU – never forwarded to PCH.
FED5 0000 - FED5 FFFF	Intel® CSME	Always enabled
FED7 0000 - FED7 4FFF	Internal Device	Security feature related

Table 4-4. PCH Memory Decode Ranges (Processor Perspective) (Sheet 3 of 3)

Memory Range	Target	Dependency/Comments
128 KB anywhere in 4 GB range	LAN Controller (CSR registers)	Enable via standard PCI mechanism (Device 31:Function 6)
4 KB anywhere in 4 GB range	LAN Controller (LAN space on Flash)	Enable via standard PCI mechanism (Device 31:Function 6)
64 KB anywhere in 64-bit address range	USB Host Controller	Enable via standard PCI mechanism (Device 20, Function 0)
2 MB anywhere in 4 GB range	USB Device Controller	Enable via standard PCI mechanism (Device 20, Function 1)
24 KB anywhere in 4 GB range	USB Device Controller	Enable via standard PCI mechanism (Device 20, Function 1)
16 KB anywhere in 64-bit addressing space	Intel® HD Audio Subsystem	Enable via standard PCI mechanism (Device 31, Function 3)
4 KB anywhere in 64-bit addressing space	Intel® HD Audio Subsystem	Enable via standard PCI mechanism (Device 31, Function 3)
64 KB anywhere in 64-bit addressing space	Intel® HD Audio Subsystem	Enable via standard PCI mechanism (Device 31, Function 3)
32 Bytes anywhere in 64-bit address range	SMBus	Enable via standard PCI mechanism (Device 31: Function 4)
2 KB anywhere above 64 KB to 4 GB range	SATA Host Controller	AHCI memory-mapped registers. Enable via standard PCI mechanism (Device 23: Function 0)
Memory Base/Limit anywhere in 4 GB range	PCI Express* Root Ports 1-20	Enable via standard PCI mechanism
Prefetchable Memory Base/Limit anywhere in 64-bit address range	PCI Express Root Ports 1-16	Enable via standard PCI mechanism
16 Bytes anywhere in 64-bit address range	Intel® CSMEI #1, #2, #3, #4	Enable via standard PCI mechanism
4 KB anywhere in 4 GB range	Intel® AMT Keyboard and Text	Enable via standard PCI mechanism (Device 22: Function 3)
16 MB anywhere in 64-bit address range	P2SB	Enable via standard PCI mechanism
Eight 4 KB slots anywhere in 64-bit address range	UART, GPI and I^2C controllers	Enable via standard PCI mechanism
4 KB slots anywhere in 64-bit address range	eMMC and SDXC controllers	Enable via standard PCI mechanism
1 MB (BAR0) or 4 KB (BAR1) in 4GB range	Integrated Sensor Hub	Enable via standard PCI mechanism (Device 19: Function 0)
8 KB slot anywhere in 4 GB range	Integrated Wi-Fi*	Enable via standard PCI mechanism
8 KB slot and 4 KB slot anywhere in 4 GB range	PMC	Enable via standard PCI mechanism
8 KB slot and 4 KB slot anywhere in 4 GB range	Shared SRAM	Enable via standard PCI mechanism

4.2.1 Boot Block Update Scheme

The PCH supports a "Top-Block Swap" mode that has the PCH swap the top block in the FWH or SPI flash (the boot block) with another location. This allows for safe update of the Boot Block (even if a power failure occurs). When the "top-swap" enable bit is set, the PCH will invert A16 for cycles going to the upper two 64-KB blocks in the FWH or appropriate address lines as selected in Boot Block Size (BOOT_BLOCK_SIZE) soft strap for SPI.

For FHW when top swap is enabled, accesses to FFFF_0000h-FFFF_FFFFh are directed to FFFE_0000h-FFFE_FFFFh and vice versa. When the Top Swap Enable bit is 0, the PCH will not invert A16.

For SPI when top swap is enabled, the behavior is as described below. When the Top Swap Enable bit is 0, the PCH will not invert any address bit.

Table 4-5. Boot Block Update Scheme

BOOT_BLOCK_SIZE Value	Accesses To	Being Directed To
000 (64KB)	FFFF_0000h - FFFF_FFFFh	FFFE_0000h - FFFE_FFFFh and vice versa
001 (128KB)	FFFE_0000h - FFFF_FFFFh	FFFC_0000h - FFFD_FFFFh and vice versa
010 (256KB)	FFFC_0000h - FFFF_FFFFh	FFF8_0000h - FFFB_FFFFh and vice versa
011 (512KB)	FFF8_0000h - FFFF_FFFFh	FFF0_0000h - FFF7_FFFFh and vice versa
100 (1MB)	FFF0_0000h - FFFF_FFFFh	FFE0_0000h - FFEF_FFFFh and vice versa
101 - 111	Reserved	Reserved
Note: This bit is automatically set to 0 by RTEST#, but not by PLTRST#.		

The scheme is based on the concept that the top block is reserved as the “boot” block, and the block immediately below the top block is reserved for doing boot-block updates.

The algorithm is:

1. Software copies the top block to the block immediately below the top
2. Software checks that the copied block is correct. This could be done by performing a checksum calculation.
3. Software sets the “Top-Block Swap” bit. This will invert the appropriate address bits for the cycles going to the FWH or the SPI.
4. Software erases the top block.
5. Software writes the new top block.
6. Software checks the new top block.
7. Software clears the top-block swap bit.
8. Software sets the Top_Swap Lock-Down bit.

If a power failure occurs at any point after step 3, the system will be able to boot from the copy of the boot block that is stored in the block below the top. This is because the top-swap bit is backed in the RTC well.

There is one remaining unusual case that could occur if the RTC battery is not sufficiently high to maintain the RTC well. To avoid the potentially fatal case (where the Top-Swap bit is NOT set, but the top block is not valid), a pin strap will allow forcing the top-swap bit to be set. This would be a last resort to allow the user to get the system to boot (and avoid having to de-solder the system flash).

When the top-swap strap is used, the top-swap bit will be forced to 1 (cannot be cleared by software).

The algorithm to put in the BIOS spec is as follows:

1. If an RTC well power failure is experienced during a boot block update, the system will probably not be able to boot at that point.

2. The user can set the Top-Swap pin strap and force the system to boot from the 2nd block. The code in the 2nd block should read the valid BIOS image from disk (probably a floppy or CD-ROM) and put it into the top-swap.
3. The BIOS will not be able to clear the Top-Swap bit (because the jumper is in place). The user should then remove the jumper and reboot.

§ §

5 System Management

5.1 Acronyms

Acronyms	Description
BMC	Baseboard Management Controller
NFC	Near-Field Communication
SPD	Serial Presence Detect
TCO	Total Cost of Ownership

5.2 Feature Overview

The PCH provides various functions to make a system easier to manage and to lower the Total Cost of Ownership (TCO) of the system. Features and functions can be augmented using external A/D converters and GPIOs, as well as an external micro controller.

The following features and functions are supported by the PCH:

- First timer timeout to generate SMI# after programmable time:
  - The first timer timeout causes an SMI#, allowing SMM-based recovery from OS lock up
- Second hard-coded timer timeout to generate reboot:
  - This second timer is used only after the 1st timeout occurs
  - The second timeout allows for automatic system reset and reboot if a HW error is detected
  - Option to prevent reset the second timeout via HW strap
- Various Error detection (such as ECC Errors) indicated by host controller:
  - Can generate SMI#, SCI, SERR, SMI, or TCO interrupt

5.3 Theory of Operation

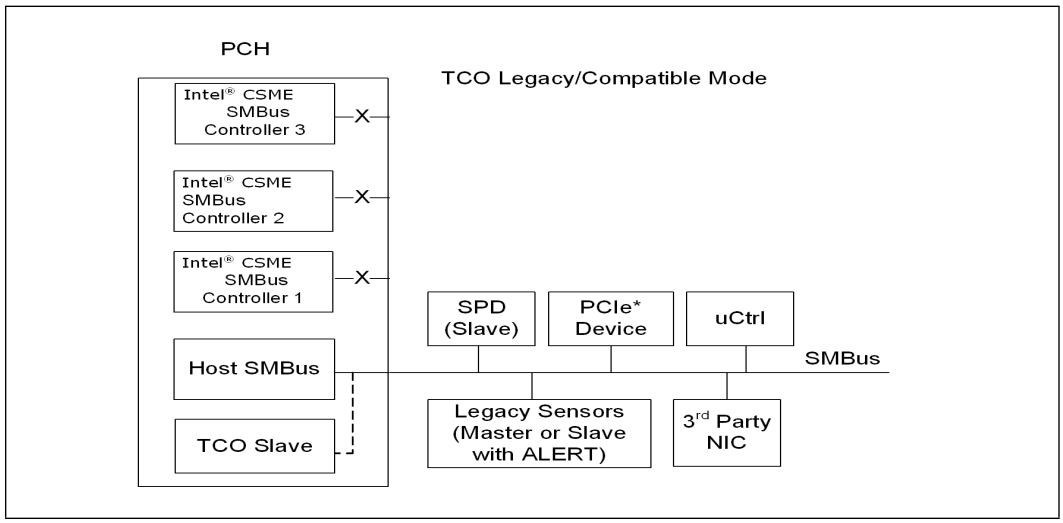
The System Management functions are designed to allow the system to diagnose failing subsystems. The intent of this logic is that some of the system management functionality can be provided without the aid of an external microcontroller.

5.3.1 TCO Modes

5.3.1.1 TCO Compatible Mode

In TCO Legacy/Compatible mode, only the host SMBus is used. The TCO Slave is connected to the host SMBus internally by default. In this mode, the Intel® Management Engine (Intel® CSME) SMBus controllers are not used and should be disabled by soft strap.

Figure 5-1. TCO Compatible Mode SMBus Configuration



In TCO Legacy/Compatible mode the PCH can function directly with an external LAN controller or equivalent external LAN controller to report messages to a network management console without the aid of the system processor. This is crucial in cases where the processor is malfunctioning or cannot function due to being in a low-power state. Below table includes a list of events that will report messages to the network management console.

Table 5-1. Event Transitions that Cause Messages

Event	Assertion?	Deassertion?	Comments
Watchdog Timer Expired	Yes	NA	"Hung S0" state entered
SMBALERT# pin	Yes	Yes	Must be in "Hung S0" state
BATLOW#	Yes	Yes	Must be in "Hung S0" state
CPU_PWR_FLR	Yes	No	"Hung S0" state entered

5.3.1.2 Advanced TCO Mode

The PCH supports the Advanced TCO mode in which SMLink0 and SMLink1 are used in addition to the host SMBus.

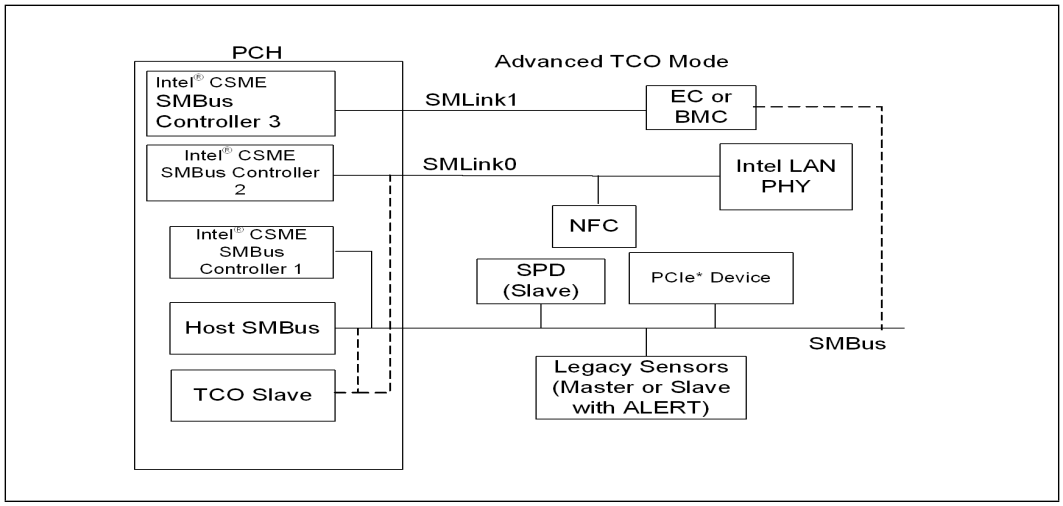
In this mode, the Intel® CSME SMBus controllers must be enabled by soft strap in the flash descriptor. Refer Figure 5-2 for more details.

In advanced TCO mode, the TCO slave can either be connected to the host SMBus or the SMLink0.

SMLink0 is targeted for integrated LAN and NFC use. When an Intel LAN PHY is connected to SMLink0, a soft strap must be set to indicate that the PHY is connected to SMLink0. When the Fast Mode is enabled using a soft strap, the interface will be running at the frequency of up to 1 MHz depending on different factors such as board routing or bus loading.

SMLink1 can be connected to an Embedded Controller (EC) or Baseboard Management Controller (BMC) use. In the case where a BMC is connected to SMLink1, the BMC communicates with the Intel Management Engine through the Intel® CSME SMBus connected to SMLink1. The host and TCO slave communicate with BMC through SMBus.

Figure 5-2. Advanced TCO Mode



§ §

6 High Precision Event Timer (HPET)

6.1 Overview

This function provides a set of timers that can be used by the operating system. The timers are defined such that the operating system may assign specific timers to be used directly by specific applications. Each timer can be configured to cause a separate interrupt.

The PCH provides eight timers. The timers are implemented as a single counter with a set of comparators. Each timer has its own comparator and value register. The counter increases monotonically. Each individual timer can generate an interrupt when the value in its value register matches the value in the main counter.

Timer 0 supports periodic interrupts.

The registers associated with these timers are mapped to a range in memory space (much like the I/O APIC). However, it is not implemented as a standard PCI function. The BIOS reports to the operating system the location of the register space using ACPI. The hardware can support an assignable decode space; however, BIOS sets this space prior to handing it over to the operating system. It is not expected that the operating system will move the location of these timers once it is set by BIOS.

6.1.1 Timer Accuracy

The timers are accurate over any 1-ms period to within 0.05% of the time specified in the timer resolution fields.

Within any 100-microsecond 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; thus, this represents an error of less than 0.2%.

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 uses the PCH's XTAL as its clock. The accuracy of the main counter is as accurate as the crystal that is used in the system.The XTAL clock frequency is determined by the pin strap that is sampled on RSMRST#.

6.1.2 Timer Off-load

The PCH supports a timer off-load feature that allows the HPET timers to remain operational during very low power S0 operational modes when the PCH's XTAL clock is disabled. The clock source during this off-load is the Real Time Clock's 32.768 kHz clock. This clock is calibrated against the PCH's XTAL clock during boot time to an accuracy that ensures the error introduced by this off-load is less than 10 ppb (.000001%).

When the PCH's XTAL clock is active, the 64-bit counter will increment by one each cycle of the PCH's XTAL clock when enabled. When the PCH's XTAL clock is disabled, the timer is maintained using the RTC clock. The long-term (> 1 msec) frequency drift allowed by the HPET specification is 500 ppm. The off-load mechanism ensures that it contributes < 1ppm to this, which will allow this specification to be easily met given the clock crystal accuracies required for other reasons.

Timer off-load is prevented when there are HPET comparators active.

The HPET timer in the PCH runs typically on the PCH's XTAL crystal clock and is off-loaded to the 32 kHz clock once the processor enters C10. This is the state where there are no C10 wake events pending and when the off-load calibrator is not running. HPET timer re-uses this 28-bit calibration value calculated by PMC when counting on the 32-kHz clock. During C10 entry, PMC sends an indication to HPET to off-load and keeps the indication active as long as the processor is in C10 on the 32 kHz clock. The HPET counter will be off-loaded to the 32 kHz clock domain to allow the PCH's XTAL MHz clock to shut down when it has no active comparators.

6.1.2.1 Theory of operation

The Off-loadable Timer Block consists of a 64b fast clock counter and an 82b slow clock counter. During fast clock mode the counter increments by one on every rising edge of the fast clock. During slow clock mode, the 82-bit slow clock counter will increment by the value provided by the Off-load Calibrator.

The Off-loadable Timer will accept an input to tell it when to switch to the slow RTC clock mode and provide an indication of when it is using the slow clock mode. The switch will only take place on the slow clock rising edge, so for the 32 kHz RTC clock the maximum delay is around 30 microseconds to switch to or from slow clock mode. Both of these flags will be in the fast clock domain.

When transitioning from fast clock to slow clock, the fast clock value will be loaded into the upper 64b of the 82b counter, with the 18 LSBs set to zero. The actual transition through happens in two stages to avoid metastability. There is a fast clock sampling of the slow clock through a double flop synchronizer. Following a request to transition to the slow clock, the edge of the slow clock is detected and this causes the fast clock value to park. At this point the fast clock can be gated. On the next rising edge of the slow clock, the parked fast clock value (in the upper 64b of an 82b value) is added to the value from the Off-load Calibrator. On subsequent edges while in slow clock mode the slow clock counter increments its count by the value from the Off-load Calibrator.

When transitioning from slow clock to fast clock, the fast clock waits until it samples a rising edge of the slow clock through its synchronizer and then loads the upper 64b of the slow clock value as the fast count value. It then de-asserts the indication that slow clock mode is active. The 32 kHz clock counter no longer counts. The 64-bit MSB will be over-written when the 32 kHz counter is reloaded once conditions are met to enable the 32 kHz HPET counter but the 18-bit LSB is retained and it is not cleared out during the next reload cycle to avoid losing the fractional part of the counter.

After initiating a transition from fast clock to slow clock and parking the fast counter value, the fast counter no longer tracks. This means if a transition back to fast clock is requested before the entry into off-load slow clock mode completes, the Off-loadable Timer must wait until the next slow clock edge to restart. This case effectively performs the fast clock to slow clock and back to fast clock on the same slow clock edge.

6.1.3 Interrupt Mapping

The interrupts associated with the various timers have several interrupt mapping options. When reprogramming the HPET interrupt routing scheme (LEG_RT_CNF bit in the General Config Register), a spurious interrupt may occur. This is because the other source of the interrupt (8254 timer) may be asserted. Software should mask interrupts prior to clearing the LEG_RT_CNF bit.

6.1.3.1 Mapping Option #1 (Legacy Replacement Option)

In this case, the Legacy Replacement Rout bit (LEG_RT_CNF) is set. This forces the mapping found in Table 6-1.

Table 6-1. Legacy Replacement Routing

Timer	8259 Mapping	APIC Mapping	Comment
0	IRQ0	IRQ2	In this case, the 8254 timer will not cause any interrupts
1	IRQ8	IRQ8	In this case, the RTC will not cause any interrupts.
2 and 3	Per IRQ Routing Field.	Per IRQ Routing Field	
4, 5, 6, 7	not available	not available	
Note: The Legacy Option does not preclude delivery of IRQ0/IRQ8 using processor interrupts messages.			

6.1.3.2 Mapping Option #2 (Standard Option)

In this case, the Legacy Replacement Rout bit (LEG_RT_CNF) is 0. Each timer has its own routing control. The interrupts can be routed to various interrupts in the 8259 or I/O APIC. A capabilities field indicates which interrupts are valid options for routing. If a timer is set for edge-triggered mode, the timers should not be shared with any legacy interrupts.

For the PCH, the only supported interrupt values are as follows:

Timer 0 and 1: IRQ20, 21, 22, and 23 (I/O APIC only).

Timer 2: IRQ11 (8259 or I/O APIC) and IRQ20, 21, 22, and 23 (I/O APIC only).

Timer 3: IRQ12 (8259 or I/O APIC) and IRQ 20, 21, 22, and 23 (I/O APIC only).

Note: Interrupts from Timer 4, 5, 6, 7 can only be delivered via direct FSB interrupt messages.

Note: System architecture changes since the HPET specification 1.0 was released have made some of the terminology used obsolete. In particular the reference to a Front Side Bus (FSB) has no relevance to current platforms, as this interface is no longer in use. For consistency with the HPET specification though, the FSB and specifically the FSB Interrupt Delivery terminology has been maintained. Where the specification refers to FSB, this should be read as ‘processor message interface’; independent of the physical attach mechanism.

6.1.3.3 Mapping Option #3 (Processor Message Option)

In this case, the interrupts are mapped directly to processor messages without going to the 8259 or I/O (x) APIC. To use this mode, the interrupt must be configured to edge-triggered mode. The Tn_PROCMSG_EN_CNF bit must be set to enable this mode.

When the interrupt is delivered to the processor, the message is delivered to the address indicated in the Tn_PROCMSG_INT_ADDR field. The data value for the write cycle is specified in the Tn_PROCMSG_INT_VAL field.

Note: The FSB interrupt deliver option has HIGHER priority and is mutually exclusive to the standard interrupt delivery option. Thus, if the TIMERn_FSB_EN_CNF bit is set, the interrupts will be delivered via the FSB, rather than via the APIC or 8259.

The FSB interrupt delivery can be used even when the legacy mapping is used.

For the Intel PCH HPET implementation, the direct FSB interrupt delivery mode is supported, besides via 8259 or I/O APIC.

6.1.4 Periodic Versus Non-Periodic Modes

6.1.4.1 Non-Periodic Mode

This mode can be thought of as creating a one-shot.

When a timer is set up for non-periodic mode, it will generate an interrupt when the value in the main counter matches the value in the timer's comparator register. Another interrupt will be generated when the main counter matches the value in the timer's comparator register after a wrap around.

During run-time, the value in the timer's comparator value register will not be changed by the hardware. Software can of course change the value.

The Timer 0 Comparator Value register cannot be programmed reliably by a single 64-bit write in a 32-bit environment **except** if only the periodic rate is being changed during run-time. If the actual Timer 0 Comparator Value needs to be reinitialized, then the following software solution will always work regardless of the environment:

- Set TIMER0_VAL_SET_CNF bit
- Set the lower 32 bits of the Timer0 Comparator Value register
- Set TIMER0_VAL_SET_CNF bit
- Set the upper 32 bits of the Timer0 Comparator Value register

Timer 0 is configurable to 32- (default) or 64-bit mode, whereas Timers 1:7 only support 32-bit mode.

Warning: Software must be careful when programming the comparator registers. If the value written to the register is not sufficiently far in the future, then the counter may pass the value before it reaches the register and the interrupt will be missed. The BIOS should pass a data structure to the operating system to indicate that the operating system should not attempt to program the periodic timer to a rate faster than 5 microseconds.

All of the timers support non-periodic mode.

Refer to Section 2.3.9.2.1 of the *IA-PC HPET Specification* for more details of this mode.

6.1.4.2 Periodic Mode

When a timer is set up for periodic mode, the software writes a value in the timer's comparator value register. When the main counter value matches the value in the timer's comparator value register, an interrupt can be generated. The hardware will then automatically increase the value in the comparator value register by the last value written to that register.

To make the periodic mode work properly, the main counter is typically written with a value of 0 so that the first interrupt occurs at the right point for the comparator. If the main counter is not set to 0, interrupts may not occur as expected.

During run-time, the value in the timer's comparator value register can be read by software to find out when the next periodic interrupt will be generated (not the rate at which it generates interrupts). Software is expected to remember the last value written to the comparator's value register (the rate at which interrupts are generated).

If software wants to change the periodic rate, it should write a new value to the comparator value register. At the point when the timer's comparator indicates a match, this new value will be added to derive the next matching point.

If the software resets the main counter, the value in the comparator's value register needs to reset as well. This can be done by setting the **TIMERn_VAL_SET_CNF** bit. Again, to avoid race conditions, this should be done with the main counter halted. The following usage model is expected:

1. Software clears the ENABLE_CNF bit to prevent any interrupts
2. Software Clears the main counter by writing a value of 00h to it.
3. Software sets the TIMER0_VAL_SET_CNF bit.
4. Software writes the new value in the TIMER0_COMPARATOR_VAL register

Software sets the ENABLE_CNF bit to enable interrupts.

Note: As the timer period approaches zero, the interrupts associated with the periodic timer may not get completely serviced before the next timer match occurs. Interrupts may get lost and/or system performance may be degraded in this case.

Each timer is NOT required to support the periodic mode of operation. A capabilities bit indicates if the particular timer supports periodic mode. he reason for this is that supporting the periodic mode adds a significant amount of gates.

For the Intel PCH, only timer 0 will support the periodic mode. This saves a substantial number of gates.

6.1.5 Enabling the Timers

The BIOS or operating system PnP code should route the interrupts. This includes the Legacy Rout bit, Interrupt Rout bit (for each timer), and interrupt type (to select the edge or level type for each timer).

The Device Driver code should do the following for an available timer:

1. Set the Overall Enable bit (Offset 10h, bit 0).
2. Set the timer type field (selects one-shot or periodic).
3. Set the interrupt enable.
4. Set the comparator value.

6.1.6 Interrupt Levels

Interrupts directed to the internal 8259s are active high. Refer Section 20.7, "Functional Description" for information regarding the polarity programming of the I/O APIC for detecting internal interrupts.

If the interrupts are mapped to the 8259 or I/O APIC and set for level-triggered mode, they can be shared with legacy interrupts. They may be shared although it is unlikely for the operating system to attempt to do this.

If more than one timer is configured to share the same IRQ (using the TIMERn_INT_ROUT_CNF fields), then the software must configure the timers to level-triggered mode. Edge-triggered interrupts cannot be shared.

6.1.7 Handling Interrupts

Section 2.4.6 of the IA-PC HPET Specification describes handling interrupts.

6.1.8 Issues Related to 64-Bit Timers with 32-Bit Processors

Section 2.4.7 of the IA-PC HPET Specification describes issues related to 64-bit timers with 32-bit processors.

§ §

7 Thermal Management

7.1 PCH Thermal Sensor

The PCH incorporates an on-die Digital Thermal Sensor (DTS) for thermal management.

7.1.1 Modes of Operation

The DTS has two usages when enabled:

1. One use is to provide the temperature of the PCH in units of 1°C. There is a 9 bit field for the temperature, with a theoretical range from –256°C to +256°C. Practically the operational range for TS would be between -40° C and 110° C.
2. The second use is to allow programmed trip points to cause alerts to SW or in the extreme case shutdown. Temperature may be provided without having any SW alerts set.

There are two thermal alert capabilities. One is for the catastrophic event (thermal runaway) which results in an immediate system power down (S5 state). The other alert provides an indication to the platform that a particular temperature has been caused. This second alert needs to be routed to SMI or SCI based on SW programming.

7.1.2 Temperature Trip Point

The internal thermal sensor reports three trip points: Cool, Hot, and Catastrophic trip points in the order of increasing temperature.

Crossing the cool trip point when going from higher to lower temperature may generate an interrupt. Crossing the hot trip point going from lower to higher temp may generate an interrupt. Each trip point has control register bits to select what type of interrupt is generated.

Crossing the cool trip point while going from low to higher temperature or crossing the hot trip point while going from high to lower temperature will not cause an interrupt.

When triggered, the catastrophic trip point will transition the system to S5 unconditionally.

7.1.3 Thermal Sensor Accuracy ($T_{accuracy}$)

The PCH thermal sensor accuracy is:

- ±5 °C over the temperature range from 50 °C to 110 °C.
- ±7 °C over the temperature range from 30 °C to 50 °C.
- ±10 °C over the temperature range from -10 °C to 30 °C.

7.1.4 Thermal Reporting to an EC

To support a platform EC that is managing the system thermals, the PCH provides the ability for the EC to read the PCH temperature over SMLink1 or over eSPI interface. The EC will issue an SMBus read or eSPI OOB Channel request and receives a single byte of data, indicating a temperature between 0 °C and 254 °C, where 255 (0xFFh) indicates that the sensor is not enabled yet. The EC must be connected to SMLink1 for thermal reporting support.

Upon reset, the value driven to the EC will be 0xFF.This indicates that BIOS has not enabled the reporting yet. When the EC receives 0xFF for the temperature, it knows that the thermal sensor is not enabled and can assume that the system is in the boot phase with unknown temperature.

After the sensor is enabled, the EC will receive a value between 0x0 and 0x7F (0 °C to 127 °C). If the EC ever sees a value between 0x80 and 0xFE, that indicates an error has occurred, since the PCH should have shut down the platform before the temperature ever reached 128 °C (Catastrophic trip point will be below 128 °C). The PCH itself does not monitor the temperature and will not flag any error on the temperature value.

7.1.5 Thermal Trip Signal (PCHHOT#)

The PCH provides PCHHOT# signal to indicate that it has exceeded some temperature limit. The limit is set by BIOS. The temperature limit (programmed into the PHL register) is compared to the present temperature. If the present temperature is greater than the PHL value then the pin is asserted.

PCHHOT# is an O/D output and requires a Pull-up on the motherboard.

The PCH evaluates the temperature from the thermal sensor against the programmed temperature limit every 1 second.

§ §

8 Power and Ground Signals

This section describes the power rails and ground signals on the PCH.

Note: The Primary well is equivalent to the historical Suspend well such that the supply is on in S0, S3, S4, S5. FIVR is the new integrated VRs in the PCH. Refer to the Power Management Chapter for more details.

Table 8-1. Power Descriptions for PCH LP

Name	Description
VCCIN_AUX	**FIVR Input rail:** 1.8V
VCC_VNNEXT_1P05	**Used for FIVR PRIM_CORE bypass mode during Sx**: 1.05V
VCC_V1P05EXT_1P05	**Used for FIVR PCH IO bypass mode during Sx:** 1.05V
VCCA_CLKLDO_1P8	**Analog supply for internal clocks:** 1.8V
VCCPRIM_1P05	**1.05V Primary Well**: for CNVi and other internal I/O blocks.
VCCDSW_1P05	**Deep Sx Well**: 1.05V. This rail is generated by on die DSW low dropout (LDO) linear regulator to supply DSW core logic.
VCCPRIM_1P8	**1.8V Primary Well**.
VCCPRIM_3P3	**3.3V Primary Well**.
VCCSPI	**SPI Primary Well 3.3V or 1.8V**. If powered at 3.3V, the 3.3V supply can come from VCCPRIM_3P3 supply. If powered at 1.8V, the 1.8V supply can come from VCCPRIM_1P8 supply.
VCCPGPPR	**Audio Power 3.3V, 1.8V, or 1.5V**. If powered at 3.3V, the 3.3V supply can come from VCCPRIM_3P3 supply. If powered at 1.8V, the 1.8V supply can come from VCCPRIM_1P8 supply.
VCCDSW_3P3	**3.3V Deep Sx Well**.
VCCRTC	**RTC Well Supply**. This rail can drop to 2.0V if all other planes are off. This power is not expected to be shut off unless the RTC battery is removed or drained. ***Note:*** VCCRTC nominal voltage is 3.0V. This rail is intended to always come up first and always stay on. It should NOT be power cycled regularly on non-coin battery designs. ***Note:*** Implementation should not attempt to clear CMOS by using a jumper to pull VCCRTC low. Clearing CMOS can be done by using a jumper on RTCRST# or GPI.
VCCDPHY_1P24	**1.24V for CNVi logic**. This rail is generated internally with a LDO and needs to be routed to the motherboard so that the rail can be supplied back to the SoC.
VCCLDOSTD_0P85	This rail is generated internally and needs to be routed out to the motherboard for decoupling purpose.
VSS	**Ground**

Note: Leakage from VCC_VNNEXT_1P05 power rail may back drive the external bypass VR when it is not in use, and its output may float up as high as 1.15 V. This is an expected behavior. Designers should makes sure they select bypass VR with an OVP threshold that is above 1.15 V for all VCC_VNNEXT_1P05 voltage settings to avoid false VR shutdown.

§ §

9 Pin Straps

The following signals are used for static configuration. They are sampled at the rising edge of either DSW_PWROK, RSMRST#, or PCH_PWROK to select configuration and then revert later to their normal usage. To invoke the associated mode, the signal should meet the set up time of 1 us and hold time of 65us, with respect to the rising edge of the sampling signal.

The PCH implements soft straps, which are used to configure specific functions within the PCH and processor very early in the boot process before BIOS or software intervention. The PCH will read soft strap data out of the SPI device prior to the de-assertion of reset to both the Intel® Converged Security and Management Engine and the Host system.

Table 9-1. Pin Straps (Sheet 1 of 4)

Signal	Usage	When Sampled	Comment
GPP_B14 / SPKR / TIME_SYNC1 / GSPI0_CS1#	Top Swap Override	Rising edge of PCH_PWROK	The strap has a 20 kohm ± 30% internal pull-down. 0 = **Disable** "Top Swap" mode. (Default) 1 = **Enable** "Top Swap" mode. This inverts an address on access to SPI and firmware hub, so the processor believes it fetches the alternate boot block instead of the original boot-block. PCH will invert A16 (default) for cycles going to the upper two 64-KB blocks in the FWH or the appropriate address lines (A16, A17, or A18) as selected in Top Swap Block size soft strap. ***Notes:*** 1. The internal pull-down is disabled after PCH_PWROK is high. 2. Software will not be able to clear the Top Swap bit until the system is rebooted. 3. The status of this strap is readable using the Top Swap bit (Bus0, Device31, Function0, offset DCh, bit4). 4. This signal is in the primary well.
GPP_B18 / GSPI0_MOSI	No Reboot	Rising edge of PCH_PWROK	The strap has a 20 kohm ± 30% internal pull-down. 0 = **Disable** "No Reboot" mode. (Default) 1 = **Enable** "No Reboot" mode (PCH will disable the TCO Timer system reboot feature). This function is useful when running ITP/XDP. ***Notes:*** 1. The internal pull-down is disabled after PCH_PWROK is high. 2. This signal is in the primary well.
GPP_C2 / SMBALERT#	TLS Confi-dentiality	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0 = **Disable** Intel® CSME Crypto Transport Layer Security (TLS) cipher suite (no confidentiality). (Default) 1 = **Enable** Intel® CSME Crypto Transport Layer Security (TLS) cipher suite (with confidentiality). Must be pulled up to support Intel® AMT with TLS. ***Notes:*** 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.

Table 9-1. Pin Straps (Sheet 2 of 4)

Signal	Usage	When Sampled	Comment
GPP_C5 / SML0ALERT#	eSPI Disable	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0 = **Enable** eSPI. (Default) 1 = **Disable** eSPI. ***Notes:*** 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
SPI0_MOSI	Reserved	Rising edge of RSMRST#	External pull-up is required. Recommend 100K if pulled up to 3.3V or 75K if pulled up to 1.8V. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.
GPP_D10 / ISH_SPI_CLK / DDP3_CTRLDATA / GSPI2_CLK / TBT_LSX2_RXD	DDP3 I2C / TBT_LSX2 pin VCC configu-ration	Rising edge of RSMRST#	This strap has no internal pull-up or pull-down. 0 = DDP3 I2C / TBT_LSX2 pin at 1.8V 1 = DDP3 I2C / TBT_LSX2 pin at 3.3V ***Notes:*** 1. An external pull-up resistor is required if the pin is used as HDMI Display I^2C, instead of TBT_LSX. 2. This signal is in the primary well.
GPP_D12 / ISH_SPI_MOSI / DDP4_CTRLDATA / GSPI2_MOSI / TBT_LSX3_RXD	DDP4 I2C / TBT_LSX3 pin VCC configu-ration	Rising edge of RSMRST#	This strap has no internal pull-up or pull-down. 0 = DDP4 I2C / TBT_LSX3 pins at 1.8V 1 = DDP4 I2C / TBT_LSX3 pins at 3.3V ***Notes:*** 1. An external pull-up resistor is required if the pin is used as HDMI Display I^2C, instead of TBT_LSX. 2. This signal is in the primary well.
GPP_B23 / SML1ALERT# / PCHHOT# / GSPI1_CS1#	CPUNSSC Clock Frequency	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0 = 38.4 MHz clock (direct from crystal) (default) 1 = 19.2 MHz clock (derived from 38.4 MHz crystal) ***Notes:*** 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. When used as PCHHOT# and strap low, a 150K pull-up is needed to ensure it does not override the internal pull-down strap sampling. 3. This signal is in the primary well.
SPI0_IO2	Reserved	Rising edge of RSMRST#	External pull-up is required. Recommend 100K if pulled up to 3.3V or 75K if pulled up to 1.8V. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.
SPI0_IO3	Reserved	Rising edge of RSMRST#	External pull-up is required. Recommend 100K if pulled up to 3.3V or 75K if pulled up to 1.8V. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.
GPP_R2 / HDA_SDO / I2S0_TXD	Flash Descriptor Security Override	Rising edge of PCH_PWROK	This strap has a 20 kohm ± 30% internal pull-down. 0 = **Enable** security measures defined in the Flash Descriptor. (Default) 1 = **Disable** Flash Descriptor Security (override). This strap should only be asserted high using external Pull-up in manufacturing/debug environments ONLY. ***Notes:*** 1. The internal pull-down is disabled after PCH_PWROK is high. 2. This signal is in the primary well.

Table 9-1. Pin Straps (Sheet 3 of 4)

Signal	Usage	When Sampled	Comment
GPP_E6	Reserved	Rising edge of RSMRST#	External pull-up is required. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.
GPP_E19 / DDP1_CTRLDATA / TBT_LSX0_RXD	DDP1 I2C / TBT_LSX0 pins VCC configu-ration	Rising edge of RSMRST#	This strap has no internal pull-up or pull-down. 0 = DDP1 I2C / TBT_LSX0 pins at 1.8V 1 = DDP1 I2C / TBT_LSX0 pins at 3.3V ***Notes:*** 1. An external pull-up resistor is required if the pin is used as HDMI Display I^2C, instead of TBT_LSX. 2. This signal is in the primary well.
GPP_E21 / DDP2_CTRLDATA / TBT_LSX1_RXD	DDP2 I2C / TBT_LSX1 pins VCC configu-ration	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0 = DDP2 I2C / TBT_LSX1 pins at 1.8V 1 = DDP2 I2C / TBT_LSX1 pins at 3.3V ***Notes:*** 1. An external pull-up resistor is required if the pin is used as HDMI Display I^2C, instead of TBT_LSX. 2. The internal pull-down is disabled after RSMRST# de-asserts. 3. This signal is in the primary well.
DBG_PMODE	Reserved	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-up. This strap should sample high. There should NOT be any on-board device driving it to opposite direction during strap sampling. ***Notes:*** 1. The internal pull-up is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
GPD7	Reserved	Rising edge of DSW_PWROK	This strap has a 20 kohm ± 30% internal pull-down. This strap should sample LOW. There should NOT be any on-board device driving it to opposite direction during strap sampling. ***Notes:*** 1. The internal pull-down is disabled after DSW_PWROK is high. 2. This signal is in the DSW well.
GPP_F0 / CNV_BRI_DT / UART0_RTS#	XTAL Frequency Selection	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. This strap should not be pulled high since 24 MHz crystal is not supported on the PCH. 0 = 38.4 MHz (default) 1 = 24 MHz ***Notes:*** 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
GPP_F2 / CNV_RGI_DT / UART0_TXD	M.2 CNVi Mode Select	Rising edge of RSMRST#	A weak external pull-up is required. 0 = Integrated CNVi enabled. 1 = Integrated CNVi disabled. ***Note:*** When a RF companion chip is connected to the PCH CNVi interface, the device internal pull-down resistor will pull the strap low to enable CNVi interface.

Table 9-1. Pin Straps (Sheet 4 of 4)

Signal	Usage	When Sampled	Comment
GPP_H2 / CNV_BT_I2S_SDI	eSPI Flash Sharing Mode	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0 = Master Attached Flash Sharing (MAFS) is enabled. (Default) 1 = Slave Attached Flash Sharing (SAFS) is enabled. ***Notes:*** 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
INTRUDER#	SPI Voltage Configu-ration	RTCRST#	There is no internal pull-up or pull-down on the signal. An external pull-up / pull-down is required. 0 = SPI operation voltage is 3.3V (10 kohm pull-down to GND) 1 = SPI operation voltage is 1.8V (1 Mohm pull-up to VCCRTC)
GPP_H23	eSPI Flash Sharing Mode	Rising edge of RSMRST#	This signal has a weak internal pull-down. 0 = Master Attached Flash Sharing (MAFS) enabled (Default) 1 = Slave Attached Flash Sharing (SAFS) enabled. ***Notes:*** 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well. ***Warning:*** **This strap must be configured to '0' (SAFS is disabled) if the *eSPI or LPC* strap is configured to '0' (eSPI is disabled)**

§ §

10 Electrical and Thermal Characteristics

This chapter contains the DC and AC characteristics for the PCH.

10.1 Absolute Maximum Ratings

Table 10-1. PCH Absolute Power Rail Maximum and Minimum Ratings

Voltage Rail	Minimum Limits	Maximum Limits
1.05V	-0.5V	1.4V
1.8V	-0.5V	2.3V
3.3V	-0.7V	3.7V
Note: Overshoot and undershoot voltage guidelines for I/O signals are outlined in Section 10.5.		

PCH Absolute Power Rail Maximum and Minimum Ratings specifies absolute maximum and minimum ratings. At conditions outside 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 likely either not function or its reliability will be severely degraded when returned to conditions within the functional operating condition limits.

Although the PCH contains protective circuitry to resist damage from Electrostatic Discharge (ESD), precautions should always be taken to avoid high static voltages or electric fields.

10.2 Thermal Specification

Table 10-2. Operating Junction Temperature Range

Parameter	Minimum	Maximum	Unit
Operating Junction Temperature Range	-40	110	°C

10.3 General DC Characteristics

Table 10-3. PCH-U Estimated I_{cc}[3]

Voltage Rail	Voltage (V)	S0 Iccmax Current[2] (A)	Sx Icc Idle Current[4] (mA)	Deep Sx Icc Idle Current (mA)	G3 (µA)
VCCIN_AUX	1.8	32.0	201		
VCC_VNNEXT_1P05	1.05	0.2	126		
VCC_V1P05EXT_1P05	1.05	0.2	200		
VCCA_CLKLDO_1P8	1.8	0.165	3		
VCCPDSW_3P3	3.3	0.004	1.4	1.4	
VCCPGPPR	1.8	0.005	0.1		
VCCPRIM_1P8	1.8	1.300	329.4		
VCCSPI	3.3	0.003	0.4		
VCCPRIM_3P3	3.3	0.202	1.6		
VCCRTC	3.3	0.002	0.2	0.2	6

Notes:
1. The VCC rail ICC data is taken at 3.0V while the system is in a mechanical off (G3) state at room temperature.
2. Iccmax estimates assumes 110 °C.
3. The Iccmax value is a steady state current that can happen after respective power ok has asserted (or reset signal has de-asserted).
4. Sx Icc Idle assumes PCH is idle and Intel® CSME is power gated.

Table 10-4. PCH-Y Estimated I_{cc}[3]

Voltage Rail	Voltage (V)	S0 Iccmax Current[2] (A)	Sx Icc Idle Current[4] (mA)	Deep Sx Icc Idle Current (mA)	G3 (µA)
VCCIN_AUX	1.8	22.0	45.0		
VCC_VNNEXT_1P05	1.05	0.2	32.0		
VCC_V1P05EXT_1P05	1.05	0.2	60.0		
VCCA_CLKLDO_1P8	1.8	0.165	3.0		
VCCPDSW_3P3	3.3	0.003	1.4	1.4	
VCCPGPPR	1.8	0.005	0.1		
VCCPRIM_1P8	1.8	1.300	95.4		
VCCSPI	3.3	0.003	0.1		
VCCPRIM_3P3	3.3	0.202	1.6		
VCCRTC	3.3	0.002	0.2	0.2	6

Notes:
1. The VCC rail ICC data is taken at 3.0V while the system is in a mechanical off (G3) state at room temperature.
2. Iccmax estimates assumes 110 °C.
3. The Iccmax value is a steady state current that can happen after respective power ok has asserted (or reset signal has de-asserted).
4. Sx Icc Idle assumes PCH is idle and Intel® CSME is power gated.

Table 10-5. Icc Adder Per HSIO Lane

Details	Icc (A)	
	PCH-U	PCH-Y
All HISO disabled	0.072	0.035
Each PCIe* Gen 3 Lane	0.178	0.169
Each PCIe* Gen 2 Lane	0.137	0.131
Each PCIe* Gen 1 Lane	0.108	0.102
Each USB 3.2 Gen 2x1 (10 Gb/s) Lane	0.310	0.300
Each USB 3.2 Gen 1x1 (5 Gb/s) Lane	0.140	0.134
Each SATA 6.0 Gb/s Port	0.141	0.135
Each SATA 3.0 Gb/s Port	0.111	0.105
Integrated GbE Port	0.073	0.067

Table 10-6. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 1 of 8)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
Note: **For GPIO pads (GPP) listed in the Associated Signals below, all functions that are multiplexed on GPIO pads will have the same DC characteristics as the GPIO pads. Refer to the GPIO Chapter for the multiplexed functions on a specific GPIO pad.**							
Associated Signals[1]: GPP_B0 / CORE_VID0, GPP_B1 / CORE_VID1, GPP_B2 / VRALERT#, GPP_B3 / CPU_GP2, GPP_B4 / CPU_GP3, GPP_B5 / ISH_I2C0_SDA, GPP_B6 / ISH_I2C0_SCL, GPP_B7 / ISH_I2C1_SDA, GPP_B8 / ISH_I2C1_SCL, GPP_B9 / I2C5_SDA / ISH_I2C2_SDA, GPP_B10 / I2C5_SCL / ISH_I2C2_SCL, GPP_B11 / PMCALERT#, GPP_B12 / SLP_S0#, GPP_B13 / PLTRST#, GPP_B14 / SPKR / TIME_SYNC1 / GSPI0_CS1#, GPP_B15 / GSPI0_CS0#, GPP_B16 / GSPI0_CLK, GPP_B17 / GSPI0_MISO, GPP_B18 / GSPI0_MOSI, GPP_B19 / GSPI1_CS0#, GPP_B20 / GSPI1_CLK, GPP_B21 / GSPI1_MISO, GPP_B22 / GSPI1_MOSI, GPP_B23 / SML1ALERT# / PCHHOT# / GSPI1_CS1#							
3.3V Operation							
Input	V_{IH}	Input High Voltage Threshold	0.75 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.25 x VCC	V		
	I_{IL}	Input Leakage Current	-12	12	µA		
	C_{IN}	Input Pin Capacitance		10	pF		
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	
	R_{pu}	WPU 5K/20K Resistance	1K-50% 5K-70% 20K-35%	1K+100% 5K+70% 20K+35%	Ohm	0.7 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-35%	5K+70% 20K+35%	Ohm	0.3 * VCC	
1.8V Operation							
Input	V_{IH}	Input High Voltage Threshold	0.75 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.25 x VCC	V		
	I_{IL}	Input Leakage Current	-12	12	µA		
	C_{IN}	Input Pin Capacitance		10	pF		

Table 10-6. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 2 of 8)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	
	R_{pu}	WPU 5K/20K Resistance	1K-50% 5K-70% 20K-35%	1K+100% 5K+70% 20K+35%	Ohm	0.7 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-35%	5K+70% 20K+35%	Ohm	0.3 * VCC	
Notes: 1. For GPIO supported voltages, refer to GPIO chapter.							
Notes: 1. For GPIO supported voltages, refer to GPIO chapter.							
Note: **For GPIO pads (GPP) listed in the Associated Signals below, all functions that are multiplexed on GPIO pads will have the same DC characteristics as the GPIO pads. Refer to the GPIO Chapter for the multiplexed functions on a specific GPIO pad.**							
Associated Signals[1]: GPP_A0 / ESPI_IO0, GPP_A1 / ESPI_IO1, GPP_A2 / ESPI_IO2, GPP_A3 / ESPI_IO3, GPP_A4 / ESPI_CS#, GPP_A5 / ESPI_CLK, GPP_A6 / ESPI_RESET#, GPP_A7 / I2S2_SCLK, GPP_A8 / I2S2_SFRM / CNV_RF_RESET#, GPP_A9 / I2S2_TXD / MODEM_CLKREQ, GPP_A10 / I2S2_RXD, GPP_A11 / DEVSLP2, GPP_A12 / SATAXPCIE1 / SATAGP1, GPP_A13 / SATAXPCIE2 / SATAGP2, GPP_A14 / USB_OC1# / DDSP_HPD3 / DISP_MISC3, GPP_A15 / USB_OC2# / DDSP_HPD4 / DISP_MISC4, GPP_A16 / USB_OC3#, GPP_A17 / DISP_MISCC, GPP_A18 / DDSP_HPDB / DISP_MISCB, GPP_A19 / DDSP_HPD1 / DISP_MISC1, GPP_A20 / DDSP_HPD2 / DISP_MISC2, GPP_A21, GPP_A22, GPP_A23 / I2S1_SCLK							
3.3V Operation							
Input	V_{IH}	Input High Voltage Threshold	0.70 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.30 x VCC	V		
	I_{IL}	Input Leakage Current	-14	14	µA		
	C_{IN}	Input Pin Capacitance		14	pF		
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	Only for 50 ohm mode
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	Only for 50 ohm mode
	R_{pu}	WPU 5K/20K Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm	0.3 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm	0.7 * VCC	
1.8V Operation							

Table 10-6. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 3 of 8)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
Input	V_{IH}	Input High Voltage Threshold	0.75 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.25 x VCC	V		
	I_{IL}	Input Leakage Current	-14	14	µA		
	C_{IN}	Input Pin Capacitance		14	pF		
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	Only for 50 ohm mode
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	Only for 50 ohm mode
	R_{pu}	WPU 5K/20K Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm	0.3 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm	0.7 * VCC	
Notes: 1. For GPIO supported voltages, refer to GPIO chapter.							
Note: **For GPIO pads (GPP) listed in the Associated Signals below, all functions that are multiplexed on GPIO pads will have the same DC characteristics as the GPIO pads. Refer to the GPIO Chapter for the multiplexed functions on a specific GPIO pad.**							
Associated Signals[1]: GPP_R0 / HDA_BCLK / I2S0_SCLK, GPP_R1 / HDA_SYNC / I2S0_SFRM, GPP_R2 / HDA_SDO / I2S0_TXD, GPP_R3 / HDA_SDI0 / I2S0_RXD, GPP_R4 / HDA_RST#, GPP_R5 / HDA_SDI1 / I2S1_SFRM, GPP_R6 / I2S1_TXD, GPP_R7 / I2S1_RXD, SPI0_IO2, SPI0_IO3, SPI0_MOSI, SPI0_MISO, SPI0_CS2#, SPI0_CS0#, SPI0_CS1#, SPI0_CLK							
3.3V Operation							
Input	V_{IH}	Input High Voltage Threshold	0.75 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.25 x VCC	V		
	I_{IL}	Input Leakage Current	-12	12	µA		
	C_{IN}	Input Pin Capacitance		13	pF		
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	
	R_{pu}	WPU 5K/20K Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm	0.7 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm	0.3 * VCC	
1.8V Operation							

Table 10-6. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 4 of 8)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
Input	V_{IH}	Input High Voltage Threshold	0.75 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.25 x VCC	V		
	I_{IL}	Input Leakage Current	-12	12	µA		
	C_{IN}	Input Pin Capacitance		13	pF		
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	
	R_{pu}	WPU 5K/20K Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm	0.7 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm	0.3 * VCC	
Notes: 1. For GPIO supported voltages, refer to GPIO chapter.							
Note: **For GPIO pads (GPP) listed in the Associated Signals below, all functions that are multiplexed on GPIO pads will have the same DC characteristics as the GPIO pads. Refer to the GPIO Chapter for the multiplexed functions on a specific GPIO pad.**							
Associated Signals[1]: GPD0 / BATLOW#, GPD11 / LANPHYPC /DSWLDO_MON, GPD2 / LAN_WAKE#, GPD3 / PWRBTN#, GPD4 / SLP_S3#, GPD5 / SLP_S4#, GPD6 / SLP_A#, GPD7, GPD8 / SUSCLK, GPD9 / SPL_WLAN#, GPD10 / SLP_S5#, GPD11 / LANPHYPC /DSWLDO_MON, INPUT3VSEL, SLP_LAN#, SLP_SUS#, WAKE#, DRAM_RESET#							
3.3V Operation							
Input	V_{IH}	Input High Voltage Threshold	0.75 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.25 x VCC	V		
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance		14	pF		
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	
	R_{pu}	WPU 5K/20K Resistance	1K-50% 5K-70% 20K-35%	1K+100% 5K+70% 20K+35%	Ohm	0.7 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-35%	5K+70% 20K+35%	Ohm	0.3 * VCC	
1.8V Operation							
Input	V_{IH}	Input High Voltage Threshold	0.75 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.25 x VCC	V		
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance		14	pF		

Table 10-6. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 5 of 8)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	
	R_{pu}	WPU 5K/20K Resistance	1K-50% 5K-70% 20K-35%	1K+100% 5K+70% 20K+35%	Ohm	0.7 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-35%	5K+70% 20K+35%	Ohm	0.3 * VCC	
Notes: 1. For GPIO supported voltages, refer to GPIO chapter.							
Note: **For GPIO pads (GPP) listed in the Associated Signals below, all functions that are multiplexed on GPIO pads will have the same DC characteristics as the GPIO pads. Refer to the GPIO Chapter for the multiplexed functions on a specific GPIO pad.**							
Associated Signals[1]: GPP_H0 / CNV_BT_I2S_SDO, GPP_H1 / SD_PWR_EN# / CNV_BT_I2S_SDO, GPP_H2 / CNV_BT_I2S_SDO, GPP_H3 / SX_EXIT_HOLDOFF# / CNV_BT_I2S_SDO, GPP_H4 / I2C2_SDA, GPP_H5 / I2C2_SCL, GPP_H6 / I2C3_SDA, GPP_H7 / I2C3_SCL, GPP_H8 / I2C4_SDA / CNV_MFUART2_RXD, GPP_H9 / I2C4_SCL / CNV_MFUART2_TXD, GPP_H10 / SRCCLKREQ4#, GPP_H11 / SRCCLKREQ5#, GPP_H12 / M2_SKT2_CFG0, GPP_H13 / M2_SKT2_CFG1, GPP_H14 / M2_SKT2_CFG2, GPP_H15 / M2_SKT2_CFG3, GPP_H16 / DDPB_CTRLCLK, GPP_H17 / DDPB_CTRLDATA, GPP_H18 / CPU_C10_GATE#, GPP_H19 / TIME_SYNC0, GPP_H20 / IMGCLKOUT1, GPP_H21 / IMGCLKOUT2, GPP_H22 / IMGCLKOUT3, GPP_H23 / IMGCLKOUT4, GPP_D0 / ISH_GP0, GPP_D1 / ISH_GP1, GPP_D2 / ISH_GP2, GPP_D3 / ISH_GP3, GPP_D4 / IMGCLKOUT0, GPP_D5 / SRCCLKREQ0#, GPP_D6 / SRCCLKREQ1#, GPP_D7 / SRCCLKREQ2#, GPP_D8 / SRCCLKREQ3#, GPP_D9 / ISH_SPI_CS# / DDP3_CTRLCLK / GSPI2_CS0# / TBT_LSX2_TXD, GPP_D10 / ISH_SPI_CLK / DDP3_CTRLDATA / GSPI2_CLK / TBT_LSX2_RXD, GPP_D11 / ISH_SPI_MISO / DDP4_CTRLCLK / GSPI2_MISO / TBT_LSX3_TXD, GPP_D12 / ISH_SPI_MOSI / DDP4_CTRLDATA / GSPI2_MOSI / TBT_LSX3_RXD, GPP_D13 / ISH_UART0_RXD, GPP_D14 / ISH_UART0_TXD, GPP_D15 / ISH_UART0_RTS# / GSPI2_CS1# / IMGCLKOUT5, GPP_D16 / ISH_UART0_CTS# / CNV_WCEN, GPP_D17 / ISH_GP4, GPP_D18 / ISH_GP5, GPP_D19 / I2S_MCLK, GPP_C0 / SMBCLK, GPP_C1 / SMBDATA, GPP_C2 / SMBALERT#, GPP_C3 / SML0CLK, GPP_C4 / SML0DATA, GPP_C5 / SML0ALERT#, GPP_C6 / SML1CLK / SUSWARN# / SUSPWRDNACK, GPP_C7 / SML1DATA / SUSACK#, GPP_C8 / UART0_RXD, GPP_C9 / UART0_TXD, GPP_C10 / UART0_RTS#, GPP_C11 / UART0_CTS#, GPP_C12 / UART1_RXD / ISH_UART1_RXD, GPP_C13 / UART1_TXD / ISH_UART1_TXD, GPP_C14 / UART1_RTS# / ISH_UART1_RTS#, GPP_C15 / UART1_CTS# / ISH_UART1_CTS#, GPP_C16 / I2C0_SDA, GPP_C17 / I2C0_SCL, GPP_C18 / I2C1_SDA, GPP_C19 / I2C1_SCL, GPP_C20 / UART2_RXD, GPP_C21 / UART2_TXD, GPP_C22 / UART2_RTS#, GPP_C23 / UART2_CTS#							
3.3V Operation							
Input	V_{IH}	Input High Voltage Threshold	0.75 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.25 x VCC	V		
	I_{IL}	Input Leakage Current	-12	12	µA		
	C_{IN}	Input Pin Capacitance		10	pF		
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	Only for 50 ohm mode
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	Only for 50 ohm mode
	R_{pu}	WPU 5K/20K Resistance	1K-50% 5K-70% 20K-35%	1K+100% 5K+70% 20K+35%	Ohm	0.7 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-35%	5K+70% 20K+35%	Ohm	0.3 * VCC	

Table 10-6. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 6 of 8)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
1.8V Operation							
Input	V_{IH}	Input High Voltage Threshold	0.75 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.25 x VCC	V		
	I_{IL}	Input Leakage Current	-12	12	µA		
	C_{IN}	Input Pin Capacitance		10	pF		
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	Only for 50 ohm mode
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	Only for 50 ohm mode
	R_{pu}	WPU 5K/20K Resistance	1K-50% 5K-70% 20K-35%	1K+100% 5K+70% 20K+35%	Ohm	0.7 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-35%	5K+70% 20K+35%	Ohm	0.3 * VCC	
Notes: 1. For GPIO supported voltages, refer to GPIO chapter.							
Note: **For GPIO pads (GPP) listed in the Associated Signals below, all functions that are multiplexed on GPIO pads will have the same DC characteristics as the GPIO pads. Refer to the GPIO Chapter for the multiplexed functions on a specific GPIO pad.**							
Associated Signals: SYS_PWROK, SYS_RESET#, CL_RST#							
3.3V Operation							
Input	V_{IH}	Input High Voltage Threshold	0.75 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.25 x VCC	V		
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance		14	pF		
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	Only for 50 ohm mode
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	Only for 50 ohm mode
	R_{pu}	WPU 5K/20K Resistance	1K-50% 5K-70% 20K-35%	1K+100% 5K+70% 20K+35%	Ohm	0.7 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-35%	5K+70% 20K+35%	Ohm	0.3 * VCC	

Table 10-6. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 7 of 8)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
1.8V Operation							
Input	V_{IH}	Input High Voltage Threshold	0.75 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.25 x VCC	V		
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance		14	pF		
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	Only for 50 ohm mode
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	Only for 50 ohm mode
	R_{pu}	WPU 5K/20K Resistance	1K-50% 5K-70% 20K-35%	1K+100% 5K+70% 20K+35%	Ohm	0.7 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-35%	5K+70% 20K+35%	Ohm	0.3 * VCC	
Note: **For GPIO pads (GPP) listed in the Associated Signals below, all functions that are multiplexed on GPIO pads will have the same DC characteristics as the GPIO pads. Refer to the GPIO Chapter for the multiplexed functions on a specific GPIO pad.**							
Associated Signals[1]: GPP_E0 / SATAXPCIE0 / SATAGP0, GPP_E1 / SPI1_IO2, GPP_E2 / SPI1_IO3, GPP_E3 / CPU_GP0, GPP_E4 / DEVSLP0, GPP_E5 / DEVSLP1, GPP_E6, GPP_E7 / CPU_GP1, GPP_E8 / SATALED# / SPI1_CS1#, GPP_E9 / USB_OC0#, GPP_E10 / SPI1_CS# / BK0 / SBK0, GPP_E11 / SPI1_CLK / BK1 / SBK1, GPP_E12 / SPI1_MISO / BK2 / SBK2, GPP_E13 / SPI1_MOSI / BK3 / SBK3, GPP_E14 / DPPE_HPDA / DISP_MISCA, GPP_E15 / ISH_GP6, GPP_E16 / ISH_GP7, GPP_E17, GPP_E18 / DDP1_CTRLCLK / TBT_LSX0_TXD, GPP_E19 / DDP1_CTRLDATA / TBT_LSX0_RXD, GPP_E20 / DDP2_CTRLCLK / TBT_LSX1_TXD, GPP_E21 / DDP2_CTRLDATA / TBT_LSX1_RXD, GPP_E22 / DDPA_CTRLCLK / PCIE_LNK_DOWN, GPP_E23 / DDPA_CTRLDATA / BK4 / SBK4							
3.3V Operation							
Input	V_{IH}	Input High Voltage Threshold	0.70 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.30 x VCC	V		
	I_{IL}	Input Leakage Current	-14	14	µA		
	C_{IN}	Input Pin Capacitance		14	pF		
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	Only for 50 ohm mode
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	Only for 50 ohm mode
	R_{pu}	WPU 5K/20K Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm	0.7 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm	0.3 * VCC	

Table 10-6. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 8 of 8)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
1.8V Operation							
Input	V_{IH}	Input High Voltage Threshold	0.75 x VCC		V		
	V_{IL}	Input Low Voltage Threshold		0.25 x VCC	V		
	I_{IL}	Input Leakage Current	-14	14	µA		
	C_{IN}	Input Pin Capacitance		14	pF		
Output	V_{OH}	Output High Voltage Threshold	VCC - 0.45	VCC	V	I_{OH}=3 mA	Only for 50 ohm mode
	V_{OL}	Output Low Voltage Threshold		0.45	V	I_{OL}=-3 mA	Only for 50 ohm mode
	R_{pu}	WPU 5K/20K Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm	0.3 * VCC	
	R_{pd}	WPD 5K/20K Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm	0.7 * VCC	

Notes:
1. For GPIO supported voltages, refer to GPIO chapter.

Table 10-7. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 1 of 2)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
Associated Signals: INTRUDER#, RSMRST#, PCH_PWROK, DSW_PWROK, SRTCRST#							
Input	V_{IH}	Input High Voltage Threshold	0.65 x VCCRTC	VCCRTC+0.5	V		4, 6
	V_{IL}	Input Low Voltage Threshold	-0.5	0.3 x VCCRTC	V		6
Associated Signals: RTCRST#							
Input	V_{IH}	Input High Voltage Threshold	0.75 x VCCRTC	V VCCRTC+0.5	V		4, 5, 6
	V_{IL}	Input Low Voltage Threshold	-0.5	0.4 x VCCRTC	V		6
Associated Signals: RTCX1#							
Input	V_{IH}	Input High Voltage Threshold	0.8	1.2	V		
	V_{IL}	Input Low Voltage Threshold	-0.5	0.1	V		
Associated Signals: XTAL24_IN							3
Input	V_{IH}	Input High Voltage Threshold	0.8	1.2	V		

Table 10-7. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 2 of 2)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
	V_{IL}	Input Low Voltage Threshold	-0.2	0.2	V		

Notes:
1. The V_{OH} specification does not apply to open-collector or open-drain drivers. Signals of this type must have an external Pull-up resistor, and that is what determines the high-output voltage level.
2. Input characteristics apply when a signal is configured as Input or to signals that are only Inputs. Output characteristics apply when a signal is configured as an Output or to signals that are only Outputs.
3. Vpk-pk minimum for XTAL24 = 500 mV
4. VCCRTC is the voltage applied to the VCCRTC well of the PCH. When the system is in G3 state, it is generally supplied by the coin cell battery. In S5 or greater state, it is supplied by VCCSUS3_3
5. V_{IH} min should not be used as the reference point for T200 timing. Refer T200 specification for the measurement point detail
6. These buffers have input hysteresis. V_{IH} levels are for rising edge transitions and V_{IL} levels are for falling edge transitions.

Table 10-8. Signal Characteristics (Sheet 1 of 3)

Symbol	Parameter	Minimum	Maximum	Unit	Conditions	Notes
Associated Signals: PCIe*						9, 10
Gen 1						
VTX-DIFF P-P	Differential Peak to Peak Output Voltage	0.8	1.2	V		1
VTX-DIFF P-P - Low	Low power differential Peak to Peak Output Voltage	0.4	1.2	V		
VTX_CM-ACp	TX AC Common Mode Output Voltage (2.5 GT/s)	—	20	mV		
ZTX-DIFF-DC	DC Differential TX Impedance	80	120	Ohm		
VRX-DIFF p-p	Differential Input Peak to Peak Voltage	0.12	1.2	V		1
VRX_CM-ACp	AC peak Common Mode Input Voltage	—	150	mV		
Gen 2						
VTX-DIFF P-P	Differential Peak to Peak Output Voltage	0.8	1.2	V		
VTX-DIFF P-P - Low	Low power differential Peak to Peak Output Voltage	0.4	1.2	V		
VTX_CM-Acp-p	TX AC Common Mode Output Voltage (5GT/s)	—	100	mV		
ZTX-DIFF-DC	DC Differential TX Impedance	80	120	Ohm		
VRX-DIFF p-p	Differential Input Peak to Peak Voltage	0.12	1.2	V		
VRX_CM-ACp	AC peak Common Mode Input Voltage	—	150	mV		
Gen 3						
VTX-DIFF P-P	Differential Peak to Peak Output Voltage	0.8	1.3	V		
VTX-DIFF P-P - Low	Low power differential Peak to Peak Output Voltage	0.4	1.2	V		
VTX_CM-Acp-p	TX AC Common Mode Output Voltage (5GT/s)	—	100	mV		

Table 10-8. Signal Characteristics (Sheet 2 of 3)

Symbol	Parameter	Minimum	Maximum	Unit	Conditions	Notes
ZTX-DIFF-DC	DC Differential TX Impedance	80	120	Ohm		
VRX-DIFF p-p	Differential Input Peak to Peak Voltage	Refer to Stressed Voltage Eye Parameters Table in PCIe* Gen 3 industry specifications.				
VRX_CM-ACp	AC peak Common Mode Input Voltage	—	150	mV		
Associated Signals: SATA						
VIMIN-Gen1i	Minimum Input Voltage - 1.5Gb/s internal SATA	325	—	mVdiff p-p		2
VIMAX-Gen1i	Maximum Input Voltage - 1.5Gb/s internal SATA	—	600	mVdiff p-p		2
VIMIN-Gen1m	Minimum Input Voltage - 1.5Gb/s eSATA	240	—	mVdiff p-p		2
VIMAX-Gen1m	Maximum Input Voltage - 1.5Gb/s eSATA	—	600	mVdiff p-p		2
VIMIN-Gen2i	Minimum Input Voltage - 3.0Gb/s internal SATA	275	—	mVdiff p-p		2
VIMAX-Gen2i	Maximum Input Voltage - 3.0Gb/s internal SATA	—	750	mVdiff p-p		2
VIMIN-Gen2m	Minimum Input Voltage - 3.0Gb/s eSATA	240	—	mVdiff p-p		2
VIMAX-Gen2m	Maximum Input Voltage - 3.0Gb/s eSATA	—	750	mVdiff p-p		2
VIMIN-Gen3i	Minimum Input Voltage - 6.0Gb/s internal SATA	240	—	mVdiff p-p		2
VIMAX-Gen3i	Maximum Input Voltage - 6.0Gb/s internal SATA	—	1000	mVdiff p-p		2
VOMIN-Gen1i,m	Minimum Output Voltage 1.5Gb/s internal and eSATA	400	—	mVdiff p-p		3
VOMAX-Gen1i,m	Maximum Output Voltage 1.5Gb/s internal and eSATA	—	600	mVdiff p-p		3
VOMIN-Gen2i,m	Minimum Output Voltage 3.0Gb/s internal and eSATA	400	—	mVdiff p-p		3
VOMAX-Gen2i,m	Maximum Output Voltage 3.0Gb/s internal and eSATA	—	700	mVdiff p-p		3
VOMIN-Gen3i	Minimum Output Voltage 6.0Gb/s internal SATA	200	—	mVdiff p-p		3
VOMAX-Gen3i	Maximum Output Voltage 6.0Gb/s internal SATA	—	900	mVdiff p-p		3
Associated Signals: USB 2.0						
VDI	Differential Input Sensitivity	0.2	—	V		4, 6

Table 10-8. Signal Characteristics (Sheet 3 of 3)

Symbol	Parameter	Minimum	Maximum	Unit	Conditions	Notes
VCM	Differential Common Mode Range	0.8	2.5	V		5, 6
VSE	Single-Ended Receiver Threshold	0.8	2	V		6
VCRS	Output Signal Crossover Voltage	1.3	2	V		6
VOL	Output Low Voltage	—	0.4	V	I_{OL}=5 mA	6
VOH	Output High Voltage	3.3V – 0.5	—	V	I_{OH}=-2 mA	6
VHSSQ	HS Squelch Detection Threshold	100	150	mV		7
VHSDSC	HS Disconnect Detection Threshold	525	625	mV		7
VHSCM	HS Data Signaling Common Mode Voltage Range	-50	500	mV		7
VHSOI	HS Idle Level	-10	10	mV		7
VHSOH	HS Data Signaling High	360	440	mV		7
VHSOL	HS Data Signaling Low	-10	10	mV		7
VCHIRPJ	Chirp J Level	700	1100	mV		7
VCHIRPK	Chirp K Level	-900	-500	mV		7
New: VDI VCM, VSE, VCRS, VOL, VOH are USB 2.0 FS/LS electrical characteristic.						
Associated Signals: USB 3.2						
VTX-DIFF-PP	Differential Peak to Peak Output Voltage	0.8	1.2	V		
VTX-DIFF P-P - Low	Low power differential Peak to Peak Output Voltage	0.4	1.2	V		8
VTX_CM-Acp-p	TX AC Common Mode Output Voltage (5GT/s)	—	100	mV		
ZTX-DIFF-DC	DC Differential TX Impedance	72	120	Ohm		
VRX-DIFF p-p	Differential Input Peak to Peak Voltage	0.1	1.2	V		
VRX_CM-ACp	AC peak Common Mode Input Voltage	—	150	mV		

Notes:
1. PCI Express* mVdiff p-p = 2*|PCIE[x]_TXP – PCIE[x]_TXN|; PCI Express* mVdiff p-p = 2*|CIE[x]_RXP – PCIE[x]_RXN|
2. SATA Vdiff, RX (V_{IMAX}/V_{IMIN}) is measured at the SATA connector on the receiver side (generally, the motherboard connector), where SATA mVdiff p-p = 2*|SATA[x]RXP – SATA[x]RXN|.
3. SATA Vdiff, tx (V_{OMIN}/V_{OMAX}) is measured at the SATA connector on the transmit side (generally, the motherboard connector), where SATA mVdiff p-p = 2*|SATA[x]TXP – SATA[x]TXN|
4. V_{DI} = | USBPx[P] – USBPx[N] |
5. Includes VDI range.
6. Applies to Low-Speed/Full-Speed USB.
7. Applies to High-Speed USB 2.0.
8. USB 3.2 mVdiff p-p = 2*|USB3Rp[x] – USB3Rn[x]|; USB 3.1 mVdiff p-p = 2*|USB3Tp[x] – USB3Tn[x]|
9. For PCIe*, GEN1, GEN and GEN3 correspond to the PCIe base specification revision 1, 2 and 3.
10. PCIe* specifications are also applicable to the LAN port.
11. Measurement taken from single-ended waveform on a component test board.
12. Measurement taken from differential waveform on a component test board.
13. VCross is defined as the voltage where Clock = Clock#.
14. Only applies to the differential rising edge (that is, Clock rising and Clock# falling).
15. The maximum voltage including overshoot.
16. The minimum voltage including undershoot.
17. The total variation of all VCross measurements in any particular system. Note that this is a subset of VCross MIN/MAX (VCross absolute) allowed. The intent is to limit VCross induced modulation by setting VCross_Delta to be smaller than VCross absolute.

Table 10-9. Other DC Characteristics (Sheet 1 of 2)

Symbol	Power State	Parameter	Minimum (V)	Nominal (V)	Maximum (V)	Notes
VCCIN_AUX	S0	PCH and CPU I/O, VNN FIVRs Inputs	1.710	1.800	1.890	1, 3, 9, 10
VCCIN_AUX	S0	PCH and CPU I/O, VNN FIVRs Inputs	1.567	1.650	1.732	2, 3, 9, 10
VCCIN_AUX	S0i2, S0i3	PCH and CPU I/O, VNN FIVRs Inputs	1.045	1.100	1.155	1, 2, 3, 9, 10
VCC_V1P05EXT_1P05	S0, S0i2, S0i3, Sx	PCH 1P05V FIVR bypass supply for low power platforms	0.997	1.050	1.102	3, 10
VCC_VNNEXT_1P05	S0i2, S0i3	PCH Vnn FIVR bypass supply	0.710	0.760	0.810	3, 5, 10
VCC_VNNEXT_1P05	Sx	PCH Vnn FIVR bypass supply	0.997	1.050	1.102	3, 6, 10
VCCA_CLKLDO_1P8	S0, S0i2, S0i3, Sx	Supply for Crystal clock's Internal LDO, MPHY PLL LDOs	1.710	1.800	1.890	3, 4, 10
VCCPRIM_1P8	Boot S0 S0i2, S0i3, Sx	1.8 Primary Well and FIVR Boot Rail	1.710	1.800	1.890	3, 10
VCCPRIM_3P3	S0, S0i2, S0i3, Sx	3.3 Primary Well	3.135	3.300	3.465	3, 10
VCCPSPI (3.3V)	S0	SPI Primary Well – 3.3V	3.135	3.300	3.465	3, 10
VCCPSPI (1.8V)	S0	SPI Primary Well – 1.8V	1.710	1.800	1.890	3, 10
VCCPGPPR (3.3V)	S0	Intel® HD Audio Supply Primary Well_1	3.135	3.300	3.465	3, 10
VCCPGPPR (1.8V)	S0	Intel® HD Audio Supply Primary Well_2	1.710	1.800	1.890	3, 10
VCCPGPPR (1.5V)	S0	Intel® HD Audio Supply Primary Well_3	1.425	1.500	1.575	3, 10
VCCDSW_3P3	Sx	Deep Sx Well for GPD and USB 2.0	3.135	3.300	3.465	3, 10
VCCRTC	RTC	RTC Well Supply	2.0	3.0	3.3	3, 7, 8, 10
VCCPRIM_1P05	All	Loopback 1P05 power to supply to Fuse, CNVPLL and CNVLDO	N/A	N/A	N/A	11
VCC1P05	All	Loopback 1P05 power to supply to CPU Load (PLL, ST, STG, FETs)	N/A	N/A	N/A	12
VCCLDOSTD_0P85	All	Decap connection only	N/A	N/A	N/A	13
VCCDSW_1P05	Sx	Decap connection only	N/A	N/A	N/A	13
VCCDPHY_1P24	All	Decap connection only	N/A	N/A	N/A	13

Table 10-9. Other DC Characteristics (Sheet 2 of 2)

Symbol	Power State	Parameter	Minimum (V)	Nominal (V)	Maximum (V)	Notes

Notes:
1. Applies to U platforms; at the specify power state.
2. Applies to Y platforms only; at the specify power state.
3. The I/O buffer supply voltage is measured at the PCH package pins. The tolerances shown in Table 10-9 are inclusive of all noise from DC up to 20 MHz. In testing, the voltage rails should be measured with a bandwidth limited oscilloscope that has a roll off of 3dB/decade above 20 MHz.
4. Filtered supply with LC placeholder. Install L when need arises.
5. For S0i2 and S0i3, an option to bypass FIVR with external VR to achieve maximum battery life. Voltage can be scaled down from 1.05V to 0.76V. Refer to Table 10-3 and Table 10-4 for Iccmax condition.
6. For Sx, an option to bypass FIVR with external VR to achieve maximum battery life. Voltage can not be scaled. Refer to Table 10-3 and Table 10-4.
7. Maximum Crystal ESR is 50 Kohm.
8. The initial VCCRTC voltage can exceed Vmax of 3.2V (up to 3.465V) for ~1 week without concerns about damage to the PCH.
9. Refer to Processor Datasheet for complete LL2 and LL3 impedance target requirement.
10. Refer toIntel Design Studio to generate a power map that accompanies this section.
11. This rail does not connect to external voltage regulator. It is a loop-back power rail supported by PCH itself.
12. This rail does not connect to external voltage regulator. It is a loop-back power rail supported by PCH itself to CPU load.
13. Decap connection only.

10.4 AC Characteristics

Table 10-10. PCI Express* Interface Timings (Sheet 1 of 2)

Symbol	Parameter	Minimum	Maximum	Unit	Figures	Notes
Transmitter and Receiver Timings						
UI (Gen1)	Unit Interval – PCI Express	399.88	400.12	ps		5
UI (Gen 2)	Unit Interval – PCI Express	199.9	200.1	ps		5
UI (GEN3)	Unit Interval – PCI Express	124.96	125.03	ps		
TTX-EYE (Gen 1/ Gen 2)	Minimum Transmission Eye Width	0.75	—	UI	10-1	1,2
$T_{TX-EYE-MEDIAN-to-MAX-JITTER}$ (Gen 1)	Maximum time between the jitter median and maximum deviation from the median	0.125	—	UI		1,2
$T_{TX-EYE-MEDIAN-to-MAX-JITTER}$ (Gen 2)	Maximum time between the jitter median and maximum deviation from the median	0.15	—	UI		
$T_{TX-EYE-MEDIAN-to-MAX-JITTER}$ (Gen 3)	Maximum time between the jitter median and maximum deviation from the median	0.15	—	UI		
TRX-EYE (Gen 1)	Minimum Receiver Eye Width	0.4	—	UI	10-2	3,4
TRX-EYE (Gen 2)	Minimum Receiver Eye Width	0.6	—	UI		3,4
TMin-Pulse (Gen 2)	Instantaneous Pulse Width	0.9	—	UI		

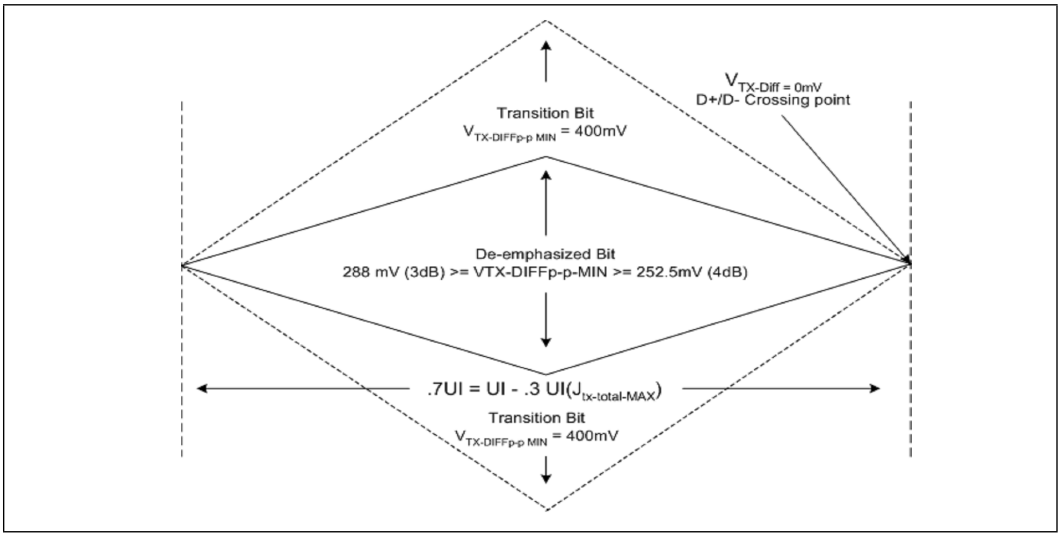
Table 10-10. PCI Express* Interface Timings (Sheet 2 of 2)

Symbol	Parameter	Minimum	Maximum	Unit	Figures	Notes

Notes: Refer to www.pcisig.com for the updated specifications.

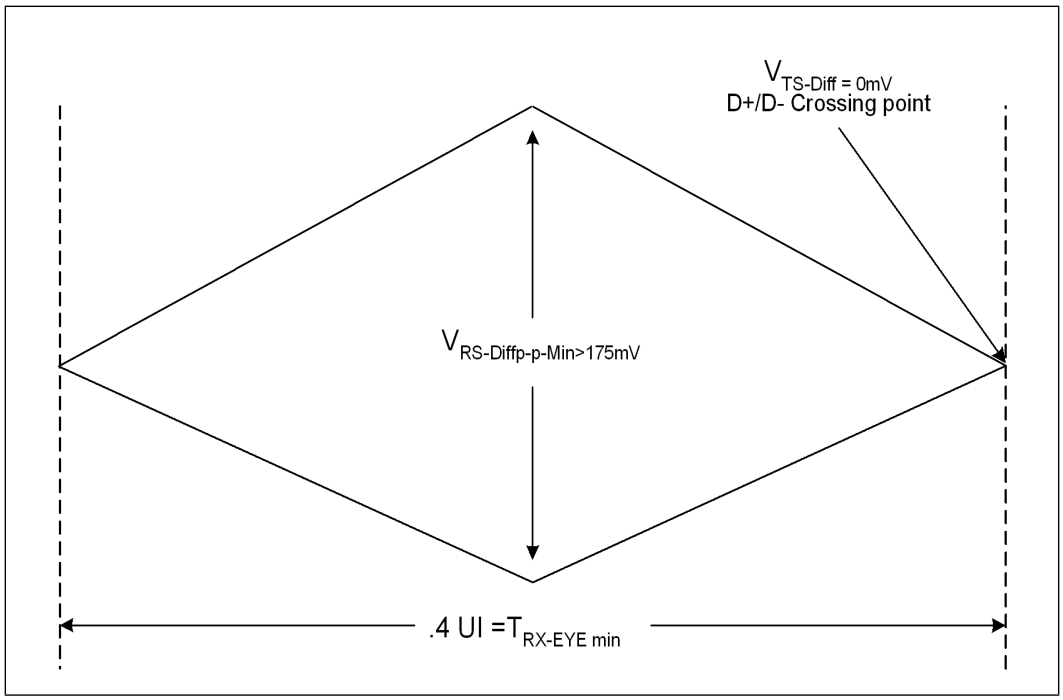
1. Specified at the measurement point into a timing and voltage compliance test load and measured over any 250 consecutive TX UIs. (also refer to the Transmitter compliance eye diagram)
2. A T_{TX-EYE} = 0.70 UI provides for a total sum of deterministic and random jitter budget of $T_{TXJITTER-MAX}$ = 0.30 UI for the Transmitter collected over any 250 consecutive TX UIs. The $T_{TXEYE-MEDIAN-to-MAX-JITTER}$ specification ensures a jitter distribution in which the median and the maximum deviation from the median is less than half of the total TX jitter budget collected over any 250 consecutive TX UIs. It should be noted that the median is not the same as the mean. The jitter median describes the point in time where the number of jitter points on either side is approximately equal as opposed to the averaged time value.
3. Specified at the measurement point and measured over any 250 consecutive UIs. The test load documented in the PCI Express* specification 2.0 should be used as the RX device when taking measurements (also refer to the Receiver compliance eye diagram). If the clocks to the RX and TX are not derived from the same reference clock, the TX UI recovered from 3500 consecutive UI must be used as a reference for the eye diagram.
4. A T_{RX-EYE} = 0.40 UI provides for a total sum of 0.60 UI deterministic and random jitter budget for the Transmitter and interconnect collected any 250 consecutive UIs. The $_{TRX-EYE-MEDIAN-to-MAX-JITTER}$ specification ensures a jitter distribution in which the median and the maximum deviation from the median is less than half of the total 0.6 UI jitter budget collected over any 250 consecutive TX UIs. It should be noted that the median is not the same as the mean. The jitter median describes the point in time where the number of jitter points on either side is approximately equal as opposed to the averaged time value. If the clocks to the RX and TX are not derived from the same reference clock, the TX UI recovered from 3500 consecutive UI must be used as the reference for the eye diagram.
5. Nominal Unit Interval is 400 ps for 2.5 GT/s and 200 ps for 5 GT/s.

Figure 10-1. PCI Express* Transmitter Eye



Note: Gen1 example is shown for the illustration. Refer to www.pcisig.com for the updated specifications.

Figure 10-2. PCI Express* Receiver Eye



Note: Gen1 example is shown for the illustration. Refer to www.pcisig.com for the updated specifications.

Table 10-11. DDC Characteristics

Signal Group: eDP_VDDEN, eDP_BKLTEN, eDP_BKLTCTL, DDP[D:C:B]_CTRLCLK, DDP[D:C:B]_CTRLDATA							
Symbol	**Parameter**	**Standard Mode**	**Fast Mode**		**1 MHz**		**Units**
		Maximum	**Minimum**	**Maximum**	**Minimum**	**Maximum**	
F_{scl}	Operating Frequency	100	0	400	0	1000	kHz
T_r	Rise Time[1]	1000	$20+0.1Cb_2$	300	—	120	ns
T_f	Fall Time[1]	300	$20+0.1Cb_2$	300	—	120	ns

Notes:
1. Measurement Point for Rise and Fall time: V_{IL}(max)–V_{IH}(min)
2. Cb = total capacitance of one bus line in pF. If mixed with High-speed mode devices, faster fall times according to High-Speed mode T_r/T_f are allowed.

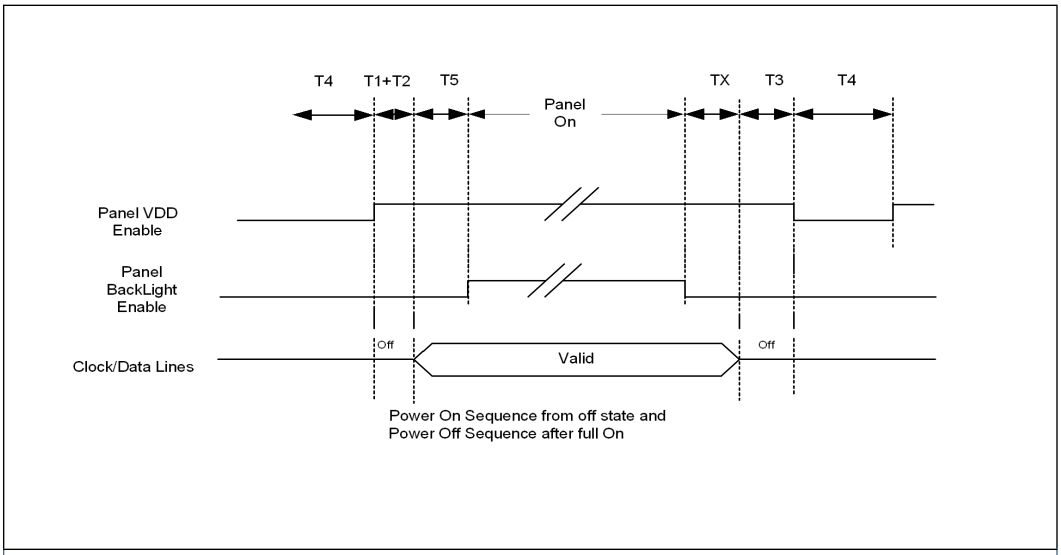
10.4.1 Panel Power Sequencing and Backlight Control

The PCH continues to integrate Panel power sequencing and Backlight control signals for eDP* interfaces on the processor.

This section provides details for the power sequence timing relationship of the panel power, the backlight enable, and the eDP* data timing delivery. To meet the panel power timing specification requirements two signals, eDP_VDDEN and eDP_BKLTEN, are provided to control the timing sequencing function of the panel and the backlight power supplies.

A defined power sequence is recommended when enabling the panel or disabling the panel. The set of timing parameters can vary from panel to panel vendor, provided that they stay within a predefined range of values. The panel VDD power, the backlight on/off state, and the eDP* data lines are all managed by an internal power sequencer.

Figure 10-3. Panel Power Sequencing



Note: Support for programming
Note: g parameters TX and T1 through T5 using software is provided.

Table 10-12. DisplayPort* Hot-Plug Detect Interface

Signal Group: DPPE_HPDA, DDSP_HPDB, DDSP_HPD1, DDSP_HPD2, DDSP_HPD3, DDSP_HPD4						
Symbol	**Parameter**	**Minimum**	**Maximum**	**Unit**	**Figures**	**Notes**
Tir	Input Time Rise	50	500	ps		
Tif	Input Time Fall	50	500	ps		
Tidr	Input Delay Rise	0.3	2.5	ns		
Tidf	Input Delay Fall	0.3	2.5	ns		

Table 10-13. Clock Timings (Sheet 1 of 3)

Symbol	Parameter	Minimum	Maximum	Unit	Notes	Figure
CLKOUT_PCIE_P/N[5:0]						
Period	Period SSC On	9.849	10.201	ns		10-5
Period	Period SSC Off	9.849	10.151	ns		10-5
DtyCyc	Duty Cycle	40	60	%		10-5
V_Swing	Differential Output Swing	300	—	mV		10-5
Slew_rise	Rising Edge Rate	1.5	4	V/ns		10-5
Slew_fall	Falling Edge Rate	1.5	4	V/ns		10-5
	Jitter	—	150	ps	8, 9, 10	
SSC	Spread Spectrum	0	0.5	%	11	
SMBus/SMLink Clock (SMBCLK, SML[1:0]CLK)						
fsmb	Operating Frequency	10	100	kHz		
t18	High Time	4	50	µs	2	10-9
t19	Low Time	4.7	—	µs		10-9
t20	Rise Time	—	1000	ns		10-9
t21	Fall Time	—	300	ns		10-9
SMLink[1,0] (SML[1:0]CLK) (Fast Mode: Refer note 15)						
fsmb	Operating Frequency	0	400	kHz		
t18_SMLFM	High Time	0.6	50	µs	2	10-9
t19_SMLFM	Low Time	1.3	—	µs		10-9
t20_SMLFM	Rise Time	—	300	ns		10-9
t21_SMLFM	Fall Time	—	300	ns		10-9
SMLink[1,0] (SML[1,0]CLK) (Fast Mode Plus: Refer note 17)						
fsmb	Operating Frequency	0	1000	kHz		
t18_SMLFMP	High Time	0.26	—	µs	2	10-9
t19_SMLFMP	Low Time	0.5	—	µs		10-9
t20_SMLFMP	Rise Time	—	120	ns		10-9
t21_SMLFMP	Fall Time	—	120	ns		10-9
I^2C Clock (Standard Mode)						
fsmb	Operating Frequency	0	100	kHz		
t18_I2CSM	High Time	4	—	µs	2	10-9
t19_I2CSM	Low Time	4.7	—	µs		10-9
t20_I2CSM	Rise Time	—	1000	ns		10-9

Table 10-13. Clock Timings (Sheet 2 of 3)

Symbol	Parameter	Minimum	Maximum	Unit	Notes	Figure
t21_I2CSM	Fall Time	—	300	ns		10-9
I^2C Clock (Fast Mode)						
fsmb	Operating Frequency	0	400	kHz		
t18_I2CFM	High Time	0.6	—	µs	2	10-9
t19_I2CFM	Low Time	1.3	—	µs		10-9
t20_I2CFM	Rise Time	20	300	ns		10-9
t21_I2CFM	Fall Time	20 x (V_{DD}/ 5.5 V)	300	ns		10-9
I^2C Clock (Fast Mode Plus)						
fsmb	Operating Frequency	0	1	MHz		
t18_I2CFMP	High Time	0.26	—	µs	2	10-9
t19_I2CFMP	Low Time	0.5	—	µs		10-9
t20_I2CFMP	Rise Time	—	120	ns		10-9
t21_I2CFMP	Fall Time	20 x (V_{DD}/ 5.5 V)	120	ns		10-9
I^2C Clock (High Speed Mode, Maximum Bus Capacitance (C_B) = 100 pF)						
fsmb	Operating Frequency	0	3.4	MHz		
t18_I2CHS1	High Time	60	—	ns	2	10-9
t19_I2CHS1	Low Time	160	—	ns		10-9
t20_I2CHS1	Rise Time	10	40	ns		10-9
t21_I2CHS1	Fall Time	10	40	ns		10-9
I^2C Clock (High Speed Mode, Maximum Bus Capacitance (C_B) = 400 pF)						
fsmb	Operating Frequency	0	1.7	MHz		
t18_I2CHS2	High Time	120	—	ns	2	10-9
t19_I2CHS2	Low Time	320	—	ns		10-9
t20_I2CHS2	Rise Time	20	80	ns		10-9
t21_I2CHS2	Fall Time	20	80	ns		10-9
HDA_BLK (Intel® High Definition Audio)						
f_{HDA}	Operating Frequency	24	—	MHz		
	Frequency Tolerance	—	100	ppm		

Table 10-13. Clock Timings (Sheet 3 of 3)

Symbol	Parameter	Minimum	Maximum	Unit	Notes	Figure
t26a	Input Jitter (refer to Clock Chip Specification)	—	300	ppm		
t27a	High Time (Measured at 0.75 Vcc)	18.75	22.91	ns		10-4
t28a	Low Time (Measured at 0.35 Vcc)	18.75	22.91	ns		10-4
Suspend Clock (SUSCLK)						
fsusclk	Operating Frequency	32		kHz	4	
t39	High Time	9.5	—	µs	4	
t39a	Low Time	9.5	—	µs	4	
XTAL_IN/XTAL_OUT						
ppm[12]	Crystal Tolerance cut accuracy maximum	35 ppm(@ 25 °C ±3 °C)				
ppm[12]	Temp Stability Maximum	30 ppm(10 – 70 °C)				
ppm[12]	Aging Maximum	5 ppm				

Notes:

1. N/A
2. The maximum high time (t18 Max.) provide a simple ensured method for devices to detect bus idle conditions.
3. BCLK Rise and Fall times are measured from 10% VDD and 90% VDD.
4. SUSCLK duty cycle can range from 30% minimum to 70% maximum.
5. Edge rates in a system as measured from 0.8 – 2.0V.
6. The active frequency can be 5 MHz, 50 MHz, or 62.5 MHz depending on the interface speed. Dynamic changes of the normal operating frequency are not allowed.
7. Testing condition: 1 KΩ Pull-up to Vcc, 1 KΩ Pull-down and 10 pF Pull-down and 1/2 inch trace.
8. Jitter is specified as cycle-to-cycle as measured between two rising edges of the clock being characterized. Period minimum and maximum includes cycle-to-cycle jitter and is also measured between two rising edges of the clock being characterized.
9. On all jitter measurements care should be taken to set the zero crossing voltage (for rising edge) of the clock to be the point where the edge rate is the fastest. Using a Math function = Average(Derivative(Ch1)) and set the averages to 64, place the cursors where the slope is the highest on the rising edge—usually this lower half of the rising edge. The reason this is defined is for users trying to measure in a system it is impossible to get the probe exactly at the end of the Transmission line with large Flip-Chip components. This results in a reflection induced ledge in the middle of the rising edge and will significantly increase measured jitter.
10. Phase jitter requirement: The designated outputs will meet the reference clock jitter requirements from the *PCI Express Base Specification*. The test is to be performed on a component test board under quiet conditions with all clock outputs on. Jitter analysis is performed using a standardized tool provided by the PCI SIG. Measurement methodology is defined in the Intel document "*PCI Express Reference Clock Jitter Measurements*. This is not for ITPXDP_P/N.
11. Spread Spectrum (SSC) is referenced to rising edge of the clock.
12. Total of crystal cut accuracy, frequency variations due to temperature, parasitics, load capacitance variations and aging is recommended to be less than 90 ppm.
13. Spread Spectrum (SSC) is referenced to rising edge of the clock.
14. Spread Spectrum (SSC) of 0.25% on CLKOUT_PCIE[7:0] and CLKOUT_PEG_[B:A] is used for WiMAX friendly clocking purposes.
15. When SMLink[1,0] is configured to run in Fast Mode (FM) using a soft strap, the supported operating range is 0 Hz ~ 400 kHz, but the typical operating frequency is in the range of 300 kHz – 400 kHz.
16. The 25 MHz output option for CLKOUTFLEX2 is derived from the 25 MHz crystal input to the PCH. The PPM of the 25 MHz output is equivalent to that of the crystal.
17. When SMLink[1,0] is configured to run in Fast Mode Plus (FMP) using a soft strap, the supported operating range is 0 Hz ~ 1 MHz, but the typical operating frequency is in the range of 900 kHz – 1000 kHz. This is the default mode for this interface.
18. Higher fall times are expected at High Speed mode. Validation data shows no functional failures with fall times as low as 9.8 ns and 8.6 ns on SDA and SCL respectively in High Speed mode at 3.3 V with Cb=100 pF.

Note: Refer to PCI Local Bus Specification for measurement details.

Figure 10-4. Clock Timing

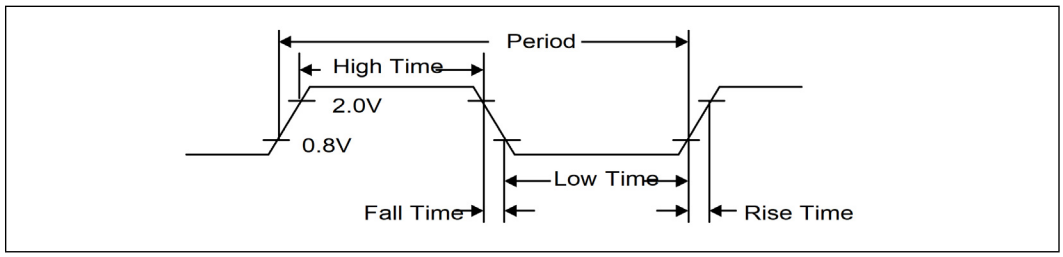


Figure 10-5. Measurement Points for Differential Waveforms

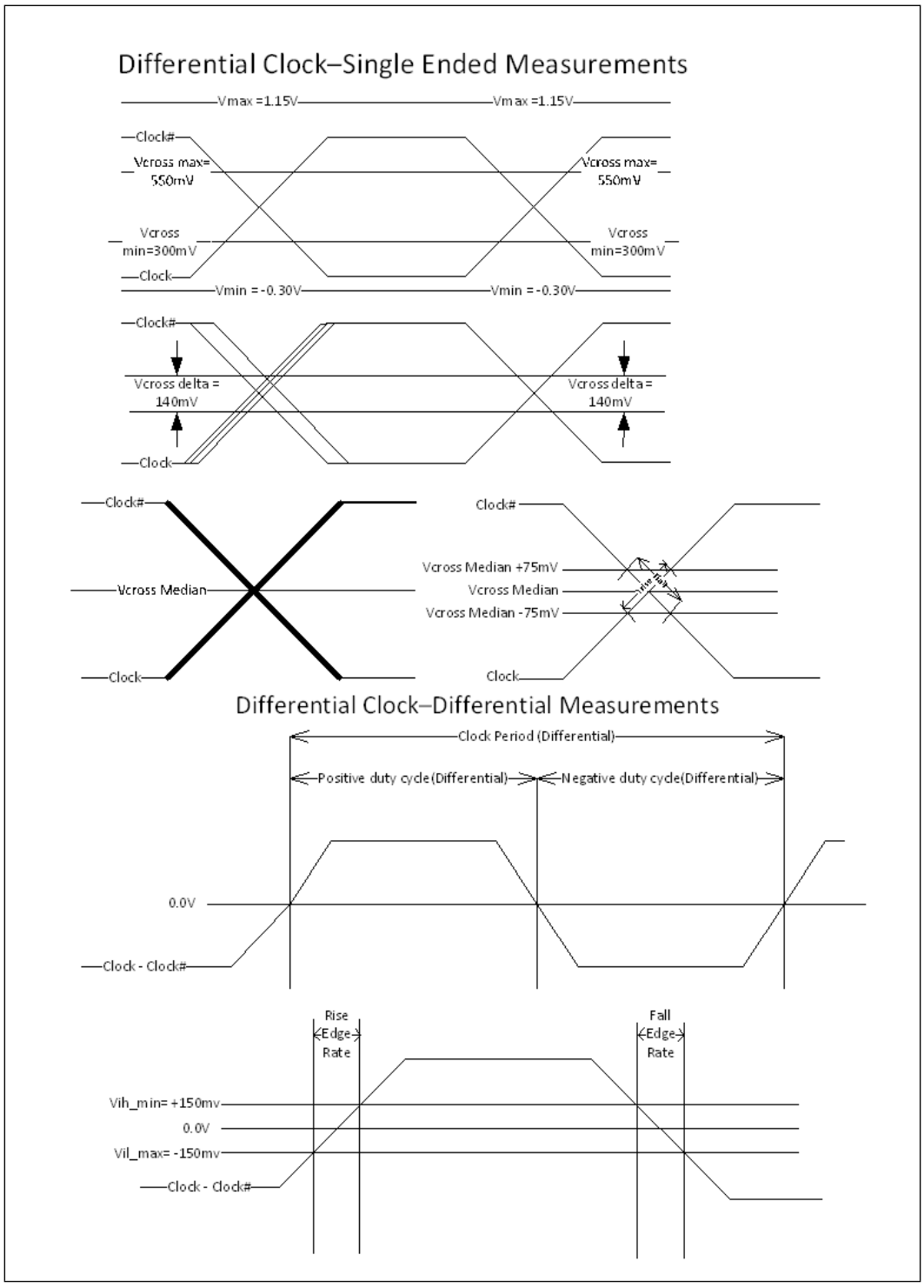


Table 10-14. USB 2.0 Timing

Sym	Parameter	Minimum	Maximum	Units	Notes	Figure
Full-speed Source (Note 7)						
t100	USBPx+, USBPx- Driver Rise Time	4	20	ns	1,6 CL = 50 pF	10-6
t101	USBPx+, USBPx- Driver Fall Time	4	20	ns	1,6 CL = 50 pF	10-6
t102	Source Differential Driver Jitter - To Next Transition - For Paired Transitions	−3.5 −4	3.5 4	ns ns	2, 3	10-7
t103	Source SE0 interval of EOP	160	175	ns	4	10-8
t104	Source Jitter for Differential Transition to SE0 Transition	−2	5	ns	5	
t105	Receiver Data Jitter Tolerance - To Next Transition - For Paired Transitions	−18.5 −9	18.5 9	ns ns	3	10-7
t106	EOP Width: Receiver must accept EOP	82	—	ns	4	10-8
t107	Width of SE0 interval during differential transition	—	14	ns		
Low-Speed Source (Note 8)						
t108	USBPx+, USBPx – Driver Rise Time	75	300	ns	1,6 CL = 200 pF CL = 600 pF	10-6
t109	USBPx+, USBPx – Driver Fall Time	75	300	ns	1,6 CL = 200 pF CL = 600 pF	10-6
t110	Source Differential Driver Jitter - To Next Transition - For Paired Transitions	−25 −14	25 14	ns ns	2,3	10-7
t111	Source SE0 interval of EOP	1.25	1.5	µs	4	10-8
t112	Source Jitter for Differential Transition to SE0 Transition	−40	100	ns	5	
t113	Receiver Data Jitter Tolerance - To Next Transition - For Paired Transitions	−152 −200	152 200	ns ns	3	10-7
t114	EOP Width: Receiver must accept EOP	670	—	ns	4	10-8
t115	Width of SE0 interval during differential transition	—	210	ns		

Notes:
1. Driver output resistance under steady state drive is specified at 28 Ω at minimum and 43 Ω at maximum.
2. Timing difference between the differential data signals.
3. Measured at crossover point of differential data signals.
4. Measured at 50% swing point of data signals.
5. Measured from last crossover point to 50% swing point of data line at leading edge of EOP.
6. Measured from 10% to 90% of the data signal.
7. Full-speed Data Rate has minimum of 11.97 Mb/s and maximum of 12.03 Mb/s.
8. Low-speed Data Rate has a minimum of 1.48 Mb/s and a maximum of 1.52 Mb/s.

Table 10-15. USB 3.2 Interface Transmit and Receiver Timings

Sym	Parameter	USB 3.2 Gen 1x1 (5 Gb/s)		USB 3.2 Gen 2x1 (10 Gb/s)		Units
		Minimum	Maximum	Minimum	Maximum	
UI	Unit Interval	199.94	200.06	99.97	100.03	ps
T_{TX-EYE}	Minimum Transmission Eye Width	0.625	—	0.646	—	UI
P_{U3}	Polling Period U3 State	—	100	—	100	mS
$P_{RX-Detect}$	Polling Period Rx Detect	—	100	—	100	mS

Figure 10-6. USB Rise and Fall Times

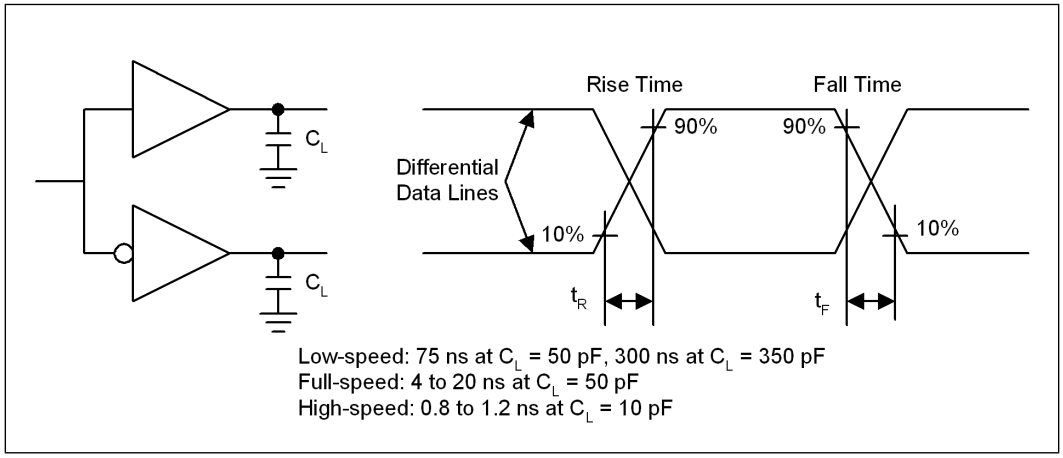


Figure 10-7. USB Jitter

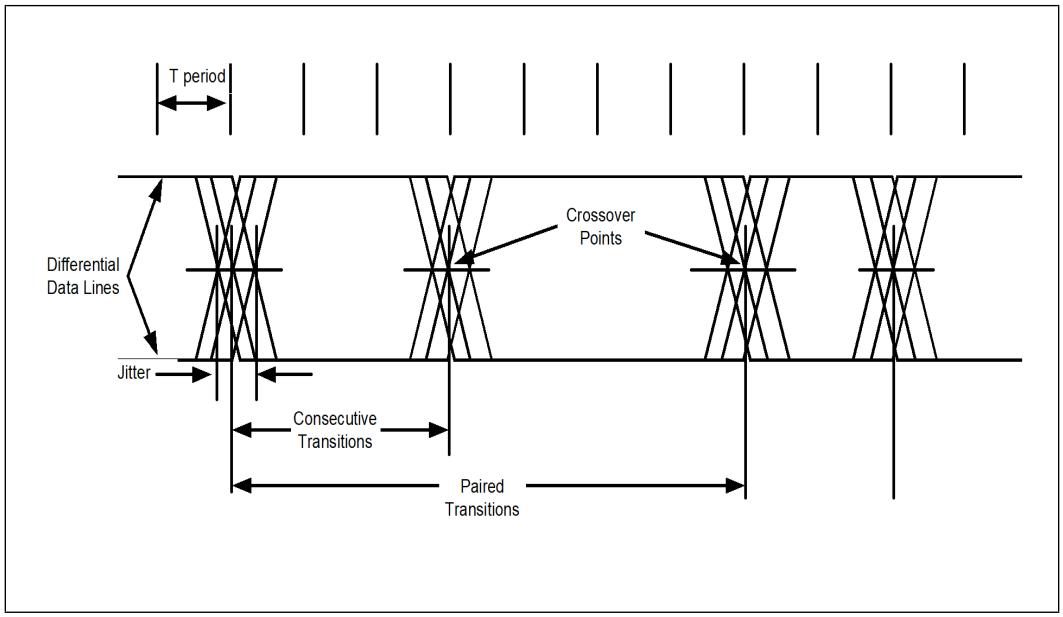


Figure 10-8. USB EOP Width

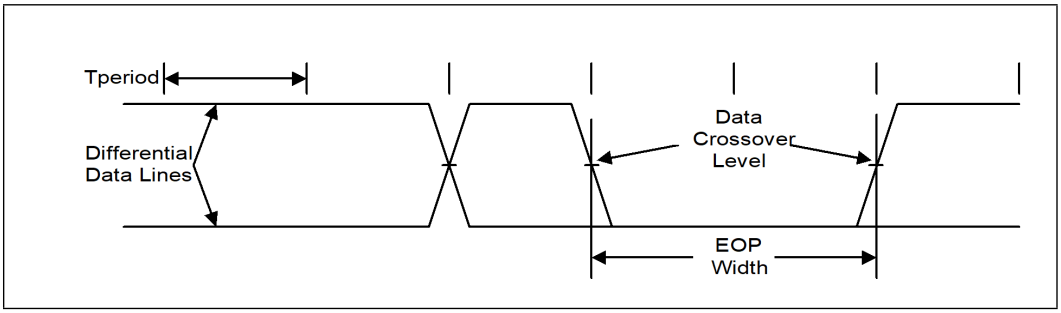


Table 10-16. SATA Interface Timings

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
UI-3	Gen III Operating Data Period (6 Gb/s)	166.6083	166.6667	ps		
t120gen3	Rise Time	0.2	0.48	UI	1	
t121gen3	Fall Time	0.2	0.48	UI	2	
t122	TX differential skew	—	20	ps		
t123	COMRESET	304	336	ns	3	
t124	COMWAKE transmit spacing	101.3	112	ns	3	
t125	OOB Operating Data period	646.67	686.67	ns	4	

Notes:
1. 20 – 80% at transmitter
2. 80 – 20% at transmitter
3. As measured from 100 mV differential crosspoints of last and first edges of burst
4. Operating data period during Out-Of-Band burst transmissions

Table 10-17. SMBus Timing (Sheet 1 of 2)

Sym	Parameter	Minimum	Maximum	Units	Notes	Figure
$t130_{100\ kHz}$	Bus Free Time Between Stop and Start Condition	4.7	—	µs		10-9
$t131_{100\ kHz}$	Hold Time after (repeated) Start Condition. After this period, the first clock is generated.	4	—	µs		10-9
$t132_{100\ kHz}$	Repeated Start Condition Setup Time	4.7	—	µs		10-9
$t133_{100\ kHz}$	Stop Condition Setup Time	4	—	µs		10-9
$t134_{100\ kHz}$	Data Hold Time	0	—	ns		10-9
$t135_{100\ kHz}$	Data Setup Time	250	—	ns		10-9
t136	Device Time Out	25	35	ms	1	
t137	Cumulative Clock Low Extend Time (slave device)	—	25	ms	2	10-10
t138	Cumulative Clock Low Extend Time (master device)	—	10	ms	3	10-10
T_{por}	Time in which a device must be operational after power-on reset	—	500	ms		

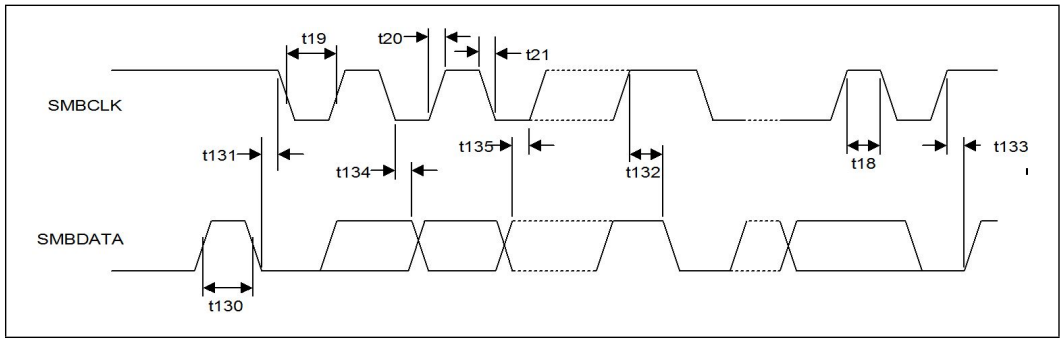
Table 10-17. SMBus Timing (Sheet 2 of 2)

Sym	Parameter	Minimum	Maximum	Units	Notes	Figure
Notes: 1. A device will timeout when any clock low exceeds this value. 2. t137 is the cumulative time a slave device is allowed to extend the clock cycles in one message from the initial start to stop. If a slave device exceeds this time, it is expected to release both its clock and data lines and reset itself. 3. t138 is the cumulative time a master device is allowed to extend its clock cycles within each byte of a message as defined from start-to-ack, ack-to-ack, or ack-to-stop.						

Table 10-18. I^2C and SMLink Timing

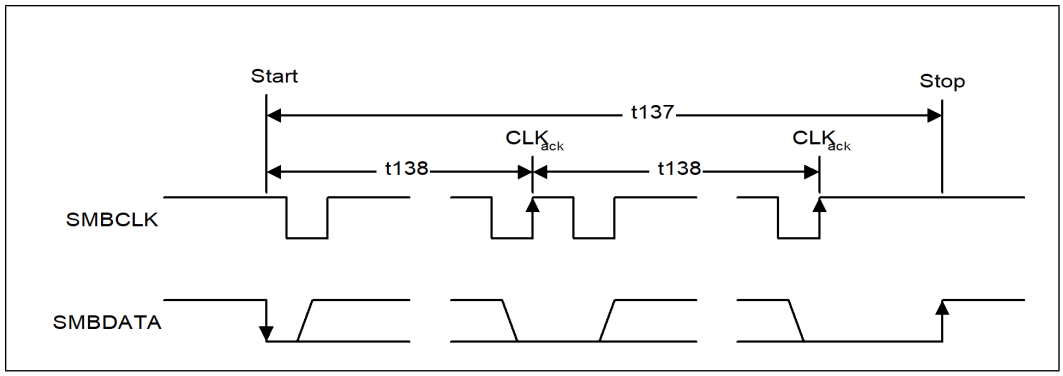
Sym[2]	Parameter	Minimum	Maximum	Units	Notes	Figure
$t130_{SM}$	Bus Free Time Between Stop and Start Condition	4.7	—	µs		10-9
$t130_{FM}$	Bus Free Time Between Stop and Start Condition	1.3	—	µs		10-9
$t130_{FMP}$	Bus Free Time Between Stop and Start Condition	0.5	—	µs		10-9
$t131_{SM}$	Hold Time after (repeated) Start Condition. After this period, the first clock is generated.	4	—	µs		10-9
$t131_{FM}$	Hold Time after (repeated) Start Condition. After this period, the first clock is generated.	0.6	—	µs		10-9
$t131_{FMP}$	Hold Time after (repeated) Start Condition. After this period, the first clock is generated.	0.26	—	µs		10-9
$t131_{HSM}$	Hold Time after (repeated) Start Condition. After this period, the first clock is generated.	160	—	ns		10-9
$t132_{SM}$	Repeated Start Condition Setup Time	4.7	—	µs		10-9
$t132_{FM}$	Repeated Start Condition Setup Time	0.6	—	µs		10-9
$t132_{FMP}$	Repeated Start Condition Setup Time	0.26	—	µs		10-9
$t132_{HSM}$	Repeated Start Condition Setup Time	160	—	ns		10-9
$t133_{SM}$	Stop Condition Setup Time	4	—	µs		10-9
$t133_{FM}$	Stop Condition Setup Time	0.6	—	µs		10-9
$t133_{FMP}$	Stop Condition Setup Time	0.26	—	µs		10-9
$t133_{HSM}$	Stop Condition Setup Time	160	—	ns		10-9
$t134_{SM}$	Data Hold Time	300	—	ns	1	10-9
$t134_{FM}$	Data Hold Time	0	—	ns		10-9
$t134_{FMP}$	Data Hold Time	0	—	ns		10-9
$t135_{SM}$	Data Setup Time	250	—	ns		10-9
$t135_{FM}$	Data Setup Time	100	—	ns		10-9
$t135_{FMP}$	Data Setup Time	50	—	ns		10-9
$t135_{HSM}$	Data Setup Time	10	—	ns		10-9
Notes: 1. t134 has a minimum timing for SMLINK is 300 ns. 2. Timings with the SM designator apply to I2C[0:5] and SMLink[1,0] when operating in Standard Mode, timings with the FM designator apply to I2C[0:5] and SMLink[1:0] when operating in Fast Mode, timings with the FMP designator apply to I2C[0:5] and SMLink[1:0] when operating in Fast Mode Plus and timing with the HSM designator apply only to I2C[0:5] when operating in High Speed Mode.						

Figure 10-9. I²C, SMLink and SMBus Transaction



Note: txx also refers to txx_SMLFM and txx_SMLFMP, txxx also refers to txxxSMLFM and txxxSMLFMP, SMBCLK also refers to SML[1:0]CLK, and SMBDATA also refers to SML[1:0]DATA.

Figure 10-10.SMBus/SMLink Timeout



Note: SMBCLK also refers to SML[1:0]CLK and SMBDATA also refers to SML[1:0]DATA in Figure 10-10.

Table 10-19. Intel® High Definition Audio (Intel® HD Audio) Timing

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t143	Time duration for which HDA_SDO is valid before HDA_BCLK edge.	6.40	13.20(24 MHz) 40.00(12 MHz)	ns		10-11
t144	Time duration for which HDA_SDO is valid after HDA_BCLK edge.	6.40	13.20(24 MHz) 40.00(12 MHz)	ns		10-11
t145	Setup time for HDA_SDI[1:0] at rising edge of HDA_BCLK	20(24 MHz) 80(12 MHz)	—	ns		10-11
t146	Hold time for HDA_SDI[1:0] at rising edge of HDA_BCLK	3	—	ns		10-11

Figure 10-11.Intel® High Definition Audio (Intel® HD Audio) Input and Output Timings

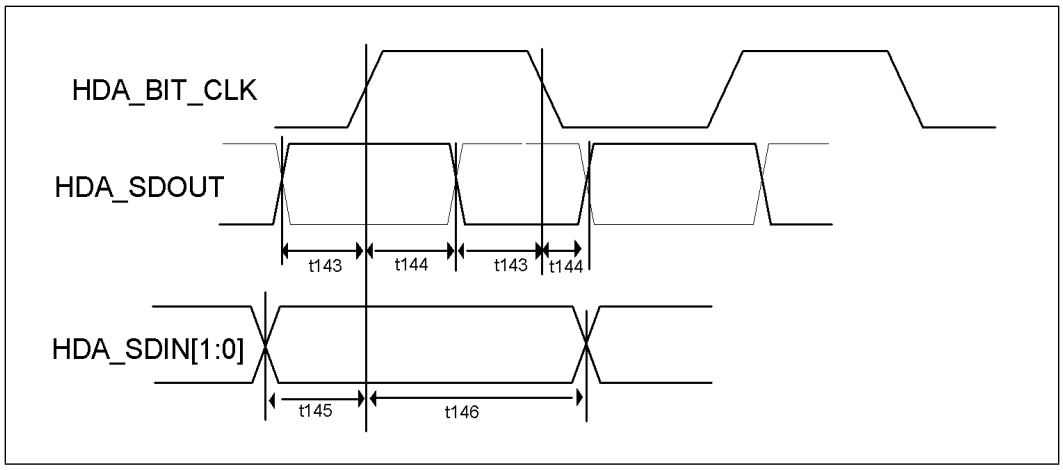


Table 10-20. DMIC Timing

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
	DMIC_DATA[0:1] Setup Time to DMIC_CLK[0:1] Rising	20	—	ns		10-13
	DMIC_DATA[0:1] Hold Time from DMIC_CLK[0:1] Rising	1	—	ns		10-13
Notes: DMIC interface rise and fall times are characterized at the PCH package ball.						

Figure 10-12. Valid Delay from Rising Clock Edge

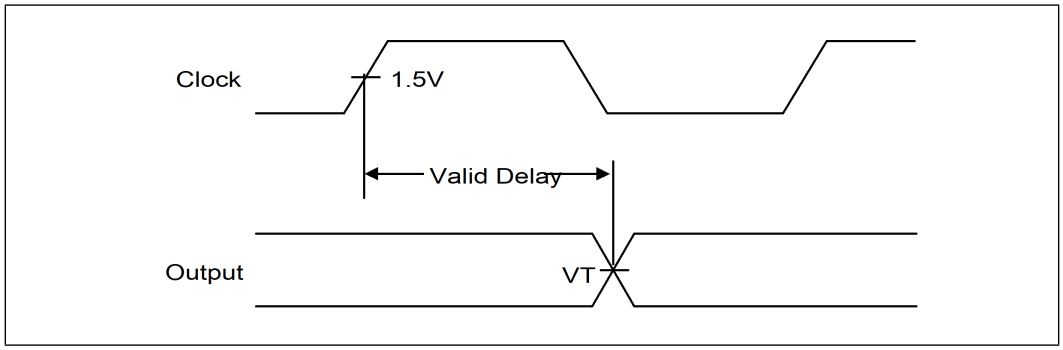


Figure 10-13.Setup and Hold Times

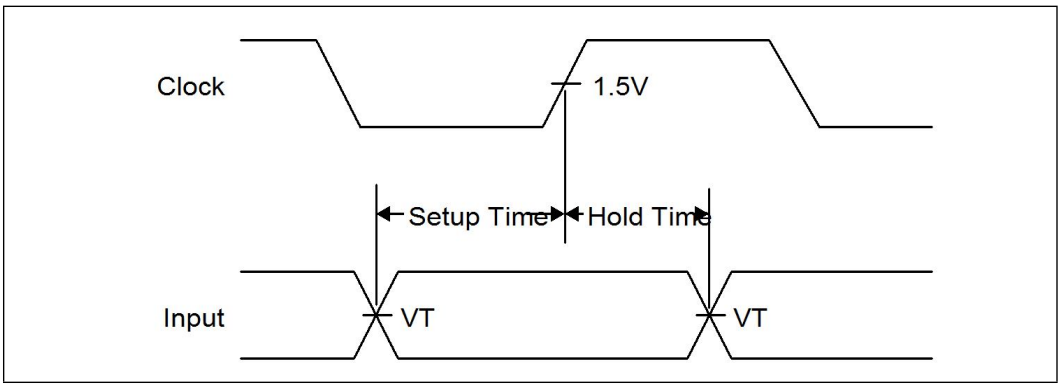


Figure 10-14.Float Delay

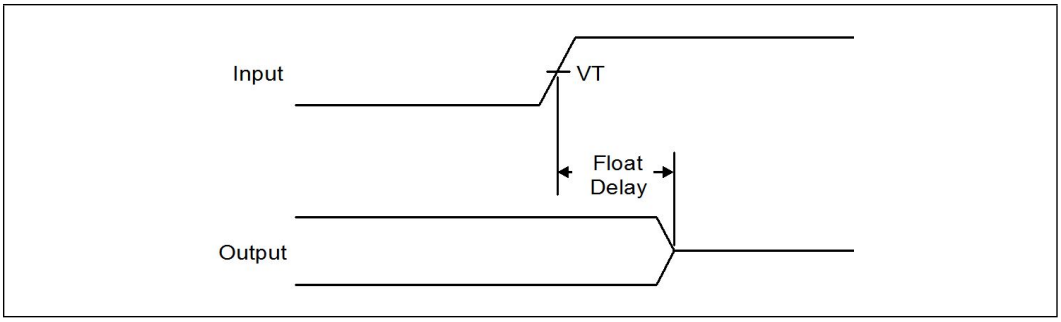


Figure 10-15.Output Enable Delay

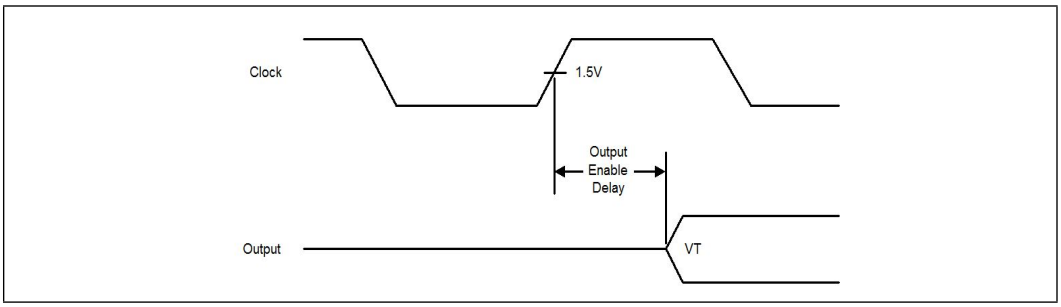


Figure 10-16.Pulse Width

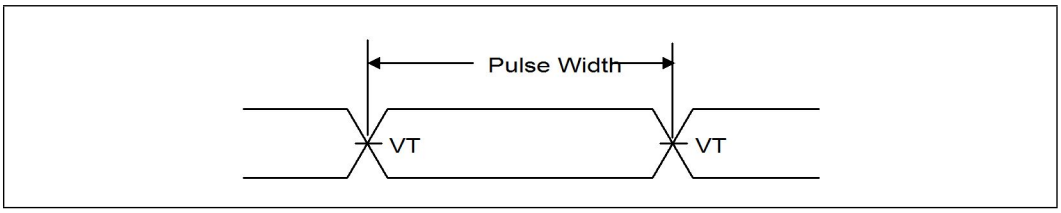


Table 10-21. Miscellaneous Timings (Sheet 1 of 2)

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t160	SERIRQ Setup Time to PCICLK Rising	7	—	ns		10-14

Table 10-21. Miscellaneous Timings (Sheet 2 of 2)

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t161	SERIRQ Hold Time from PCICLK Rising	0	—	ns		
t162	GPIO, USB Resume Pulse Width	2	—	RTCCLK		10-16
t163	SPKR Valid Delay from OSC Rising	—	200	ns		10-12

Table 10-22. SPI Timings (20 MHz)

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t180a	Serial Clock Frequency	16.8	17.48	MHz	1	
t183a	Tco of SPI MOSI and SPI I/O with respect to serial clock falling edge at the host	-13	14	ns		10-18
t184a	Setup of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	35.0	—	ns		10-18
t185a	Hold of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	1.50	—	ns		10-18
t186a	Setup of SPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		10-18
t187a	Hold of SPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		10-18
t188a	SPI CLK High time	23.84	—	ns		10-18
t189a	SPI CLK Low time	31.84	—	ns		10-18

Notes:
1. The typical clock frequency driven by the PCH is 17.14 MHz.
2. Measurement point for low time and high time is taken at 0.5(VCCSPI).
3. PCH output timing such as Tco, are simulation values, with a test load of 2pF.

Table 10-23. SPI Timings (33 MHz) (Sheet 1 of 2)

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t180b	Serial Clock Frequency	29.4	30.6	MHz	1	10-18
t183b	Tco of SPI MOSI and SPI I/O with respect to serial clock falling edge at the host	-8	8	ns		10-18
t184b	Setup of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	18.0	—	ns		10-18
t185b	Hold of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	1.50	—	ns		10-18
t186b	Setup of SPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		10-18
t187b	Hold of SPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		10-18

Table 10-23. SPI Timings (33 MHz) (Sheet 2 of 2)

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t188b	SPI CLK High time	16	—	ns		10-18
t189b	SPI CLK Low time	16	—	ns		10-18

Notes:
1. The typical clock frequency driven by the PCH is 30 MHz.
2. Measurement point for low time and high time is taken at 0.5(VCCSPI).
3. PCH output timing such as Tco, are simulation values, with a test load of 2pF.

Table 10-24. SPI Timings (50 MHz)

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t180c	Serial Clock Frequency	47.04	48.96	MHz	1	10-18
t183c	Tco of SPI MOSI and SPI I/O with respect to serial clock falling edge at the host	-3	6.2	ns		10-18
t184c	Setup of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	7.0	—	ns		10-18
t185c	Hold of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	1.50	—	ns		10-18
t186c	Setup of SPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		10-18
t187c	Hold of SPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		10-18
t188c	SPI CLK High time	7.84	—	ns	2, 3	10-18
t189c	SPI CLK Low time	11.84	—	ns	2, 3	10-18

Notes:
1. Typical clock frequency driven by the PCH is 48 MHz.
2. When using 48 MHz mode ensure target flash component can meet t188c and t189c specifications. Measurement should be taken at a point as close as possible to the package pin.
3. Measurement point for low time and high time is taken at 0.5(VCCSPI).
4. PCH output timing such as Tco, are simulation values, with a test load of 2pF.

Figure 10-17. PCH Test Load

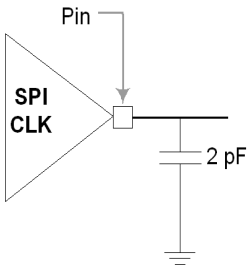


Figure 10-18.SPI Timings

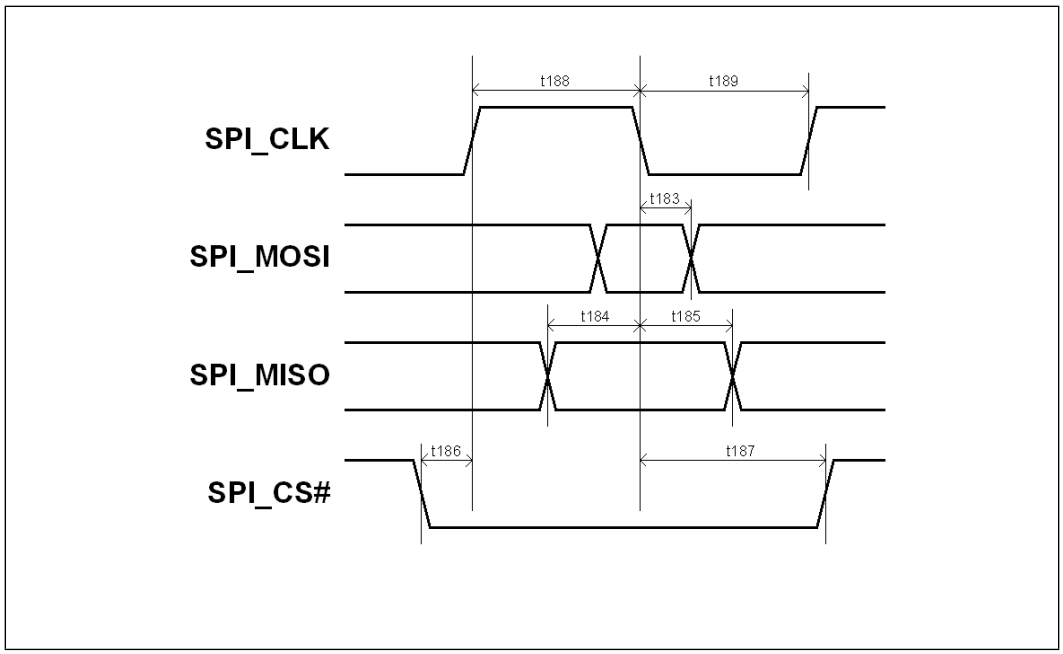


Table 10-25. GSPI Timings (25 MHz)

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
F	Serial Clock Frequency	—	25	MHz		10-19
t183	Tco of SPI MOSI with respect to serial clock falling edge	-15	7.6	ns		10-19
t184	Setup of SPI MISO and SPI I/O with respect to serial clock rising edge	3.8	—	ns		10-19
t185	Hold of SPI MISO and SPI I/O with respect to serial clock rising edge	20	—	ns		10-19
t186	Setup of SPI CS# assertion with respect to serial clock rising edge	20	—	ns		10-19
t187	Hold of SPI CS# assertion with respect to serial clock falling edge	20	—	ns		10-19

Figure 10-19.GSPI Timings

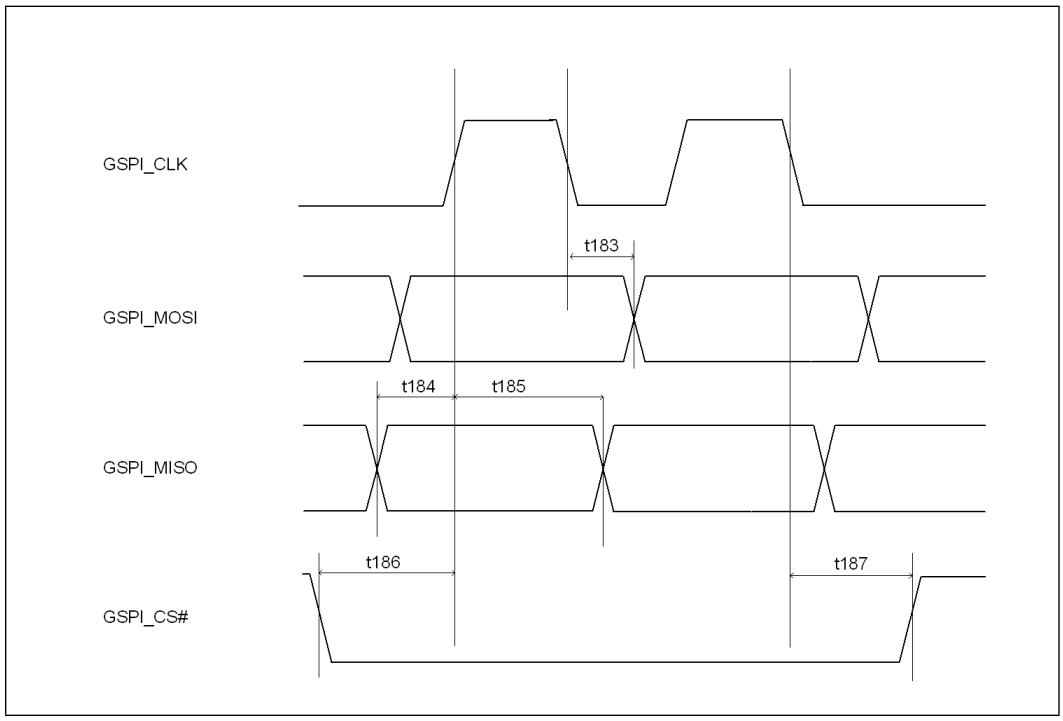


Table 10-26. Controller Link Receive Timings

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t190	Single bit time	13	—	ns		10-20
t191	Single clock period	30	—	ns		10-20
t193	Setup time before CL_CLK	0.9	—	ns		10-20
t194	Hold time after CL_CLK	0.9	—	ns		10-20
V_{IL_AC}	Input low voltage (AC)	—	CL_Vref – 0.08	V	2	
V_{IH_AC}	Input high voltage (AC)	CL_Vref +0.08	—	V	2	

Notes:
1. Measured from (CL_Vref – 50 mV to CL_Vref + 50 mV) at the receiving device side. No test load is required for this measurement as the receiving device fulfills this purpose.
2. CL_Vref = 0.12*(VccSus3_3).

Figure 10-20.Controller Link Receive Timings

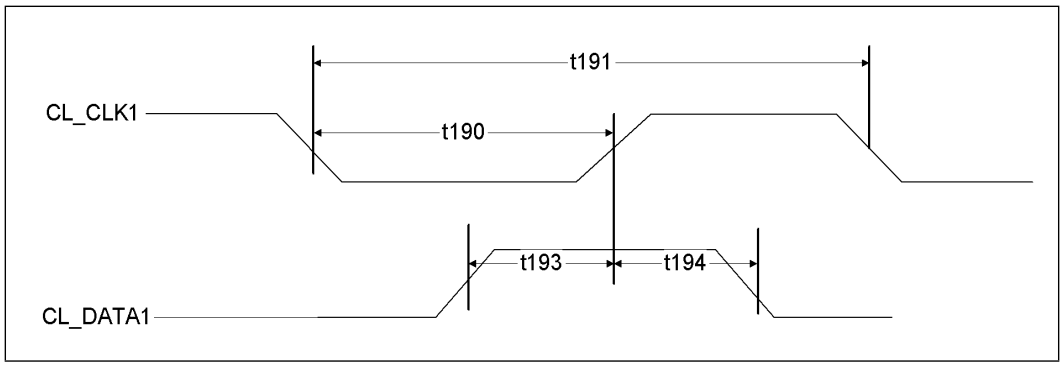


Figure 10-21.Controller Link Receive Slew Rate

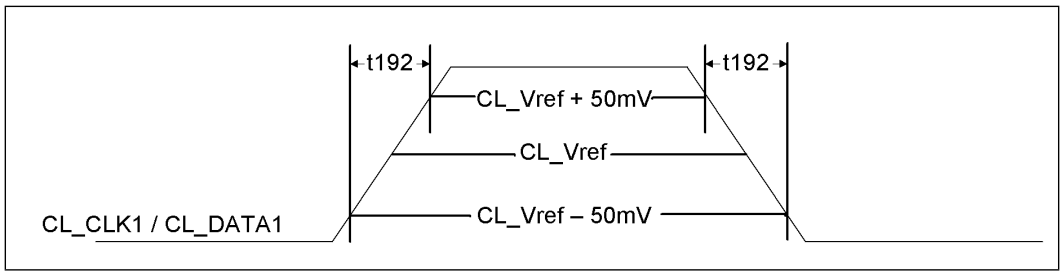


Table 10-27. UART Timings

Sym	Parameter	Minimum	Maximum	Units	Notes	Figure
F	Operating Frequency	—	6.25	MHz		
Slew_rise	Output Rise Slope	1.452	2.388	V/ns		
Slew_fall	Output Fall Slope	1.552	2.531	V/ns		

Table 10-28. I^2S Timings - Master Mode

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
SCLK						
F_{I2S}	Clock Frequency	—	12.288	MHz		
F_{I2S}	Clock Frequency	—	9.6	MHz		
SFRM						
T_{CO}	Clock to Output Delay	—8	15	ns		
RXD						
T_{SU}	Setup Time	40	—	ns		
T_{HD}	Hold Time	1	—	ns		
TXD						
T_{CO}	Clock to Output Delay	—8	15	ns		

Table 10-29. I²S Timing - Slave Mode (non S0ix)

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
SCLK						
F_{I2S}	Clock Frequency	—	12.288	MHz		
SFRM						
T_{SU}	Setup Time	9	—	ns		
T_{HD}	Hold Time	10	—	ns		
RXD						
T_{SU}	Setup Time	9	—	ns		
T_{HD}	Hold Time	10	—	ns		
TXD						
T_{CO}	Clock to Output Delay	0	21	ns		

Table 10-30. I²S Timing - Slave Mode (S0ix)

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
SCLK						
F_{I2S}	Clock Frequency	—	9.6	MHz		
SFRM						
T_{SU}	Setup Time	15	—	ns		
T_{HD}	Hold Time	10	—	ns		
RXD						
T_{SU}	Setup Time	15	—	ns		
T_{HD}	Hold Time	10	—	ns		
TXD						
T_{CO}	Clock to Output Delay	0	28	ns		

Table 10-31. ISH SPI

Parameter	Minimum	Maximum	Unit	Notes	Figure
Serial Clock Frequency	—	15	MHz		
Tco of SPI MOSI and SPI I/O with respect to serial clock falling edge at the host	-25	15	ns		
Setup of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	49	—	ns		
Hold of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	0	—	ns		

Table 10-32. eSPI Timings

Parameter	Minimum	Maximum	Units
Serial Clock Frequency	20.00	60.00	MHz
Tco of eSPI I/O with respect to serial clock falling edge at the host	-3.00(60 MHz) -9.00(30 MHz)	2.50(60 MHz) 8.00(30 MHz)	ns
Setup eSPI I/ O with respect to serial clock falling edge at the host	5.50(60 MHz) 17.00(30 MHz)		ns
Hold eSPI I/ O with respect to serial clock falling edge at the host	0.50(60 MHz) 0.50(30 MHz)		ns
eSPI Clock High time	6.2		ns
eSPI Clock Low time	6.2		ns

Notes:
1. eSPI supports 60/48/30/24/20 MHz. Duty Cycle (HIGH/LOW) for 48 MHz clock is 40/60, and same for the other frequencies are all 50/50.
2. Measurement point of low time and high time taken at 0.5(VCCSPI).

10.5 Overshoot/Undershoot Guidelines

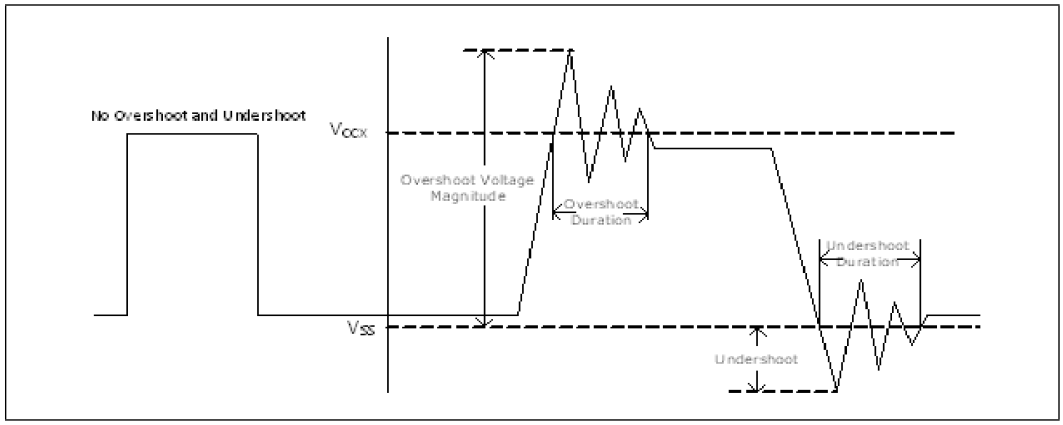
Overshoot (or undershoot) is the absolute value of the maximum voltage above VCC or below VSS. The PCH can be damaged by single and/or repeated overshoot or undershoot events on any input, output, or I/O buffer if the charge is large enough. Baseboard designs that meet signal integrity and timing requirements and that do not exceed the maximum overshoot or undershoot limits listed in Table 10-33 will ensure reliable I/O performance for the lifetime of the PCH.

Table 10-33. Overshoot/Undershoot Specifications

Signal Group	Voltage (Vccx) (V)	Overshoot Voltage Magnitude (V)	Overshoot Duration (nS)	Undershoot Voltage Magnitude (V)	Undershoot Duration (nS)
GPP_A, GPP_E, GPP_G, GPP_R, SPI	1.8	2.01	0.6	-0.15	0.6
		1.97	1.2	-0.12	1.2
	3.3	3.65	2.5	-0.26	2.5
		3.61	5	-0.21	5
GPP_F	1.8	2.27	0.6	-0.42	0.6
		2.20	1.2	-0.34	1.2
DSW, GPP_B, GPP_C, GPP_D, GPP_H, HVCMOS	1.8	1.95	0.6	-0.10	0.6
		1.90	1.2	-0.05	1.2
	3.3	3.51	2.5	-0.11	2.5
		3.45	5	-0.05	5

Notes:
1. These specifications are measured at the PCH pin.
2. Voltage (Vccx) refers to the supply voltage at the pin. Refer to Figure 10-22, "Maximum Acceptable Overshoot/Undershoot Waveform" for pictorial description of allowable overshoot/undershoot magnitude and duration.

Figure 10-22.Maximum Acceptable Overshoot/Undershoot Waveform



§ §

11 8254 Timers

11.1 Overview

The PCH contains two counters that have fixed uses. All registers and functions associated with these timers are in the Primary well. The 8254 unit is clocked by a 1.193 MHz periodic timer tick, which is functional only in S0 states. The 1.193MHz periodic timer tick is generated off the 38.4/24 MHz xtal clock.

Counter 0, System Timer

This counter functions as the system timer by controlling the state of IRQ0 and is typically 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 1 counter period after 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.

Counter 2, Speaker Tone

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 port 061h (Refer NMI Status and Control ports).

11.1.1 Timer Programming

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 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 may be written to a counter at any time without affecting the counter's 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 two-byte counts, the following precaution applies – 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 will be loaded with an incorrect count.

The Control Word Register at port 43h controls the operation of all three counters. Several commands are available:

- **Control Word Command.** Specifies which counter to read or write, the operating mode, and the count format (binary or BCD).
- **Counter Latch Command.** Latches the current count so that it can be read by the system. 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 11-1 lists the six operating modes for the interval counters.

Table 11-1. Counter Operating Modes

Mode	Function	Description
0	Out signal on end of count (=0)	Output is 0. When count goes to 0, output goes to 1 and stays at 1 until counter is reprogrammed.
1	Hardware retriggerable one-shot	Output is 0. When count goes to 0, output goes to 1 for one clock time.
2	Rate generator (divide by n counter)	Output is 1. Output goes to 0 for one clock time, then back to 1 and counter is reloaded.
3	Square wave output	Output is 1. Output goes to 0 when counter rolls over, and counter is reloaded. Output goes to 1 when counter rolls over, and counter is reloaded, and so on
4	Software triggered strobe	Output is 1. Output goes to 0 when count expires for one clock time.
5	Hardware triggered strobe	Output is 1. Output goes to 0 when count expires for one clock time.

11.1.2 Reading from the Interval Timer

It is often desirable to read the value of a counter without disturbing the count in progress. There are three methods for reading the counters—a simple read operation, counter Latch command, and the Read-Back command. Each is 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 two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other. Read, write, or programming operations for other counters may be inserted between them.

11.1.2.1 Simple Read

The first method is to perform a simple read operation. The counter is selected through Port 40h (Counter 0) 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 can be stopped by writing to the GATE bit in Port 61h.

11.1.2.2 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 is used to ensure that the count read from the counter is accurate, particularly when reading a two-byte count. The count value is then read from each counter's Count register as 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. This allows reading the contents of the counters on the fly without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one counter. Counter Latch commands do not affect the programmed mode of the counter in any way.

If a Counter is latched and then, some time later, latched again before the count is read, the second Counter Latch command is ignored. The count read is the count at the time the first Counter Latch command was issued.

11.1.2.3 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 may be used to latch multiple counter outputs at one time. This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's 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's 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 may be 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 unlatched count.

§ §

12 Audio, Voice, and Speech

12.1 Acronyms

Acronyms	Description
DMA	Direct Memory Access.
DMIC	Digital Microphone. PDM based MEMs microphone modules.
DSP	Digital Signal Processor. In AVS specifically a DSP to process audio data.
I^2S	Inter IC Sound. A serial bus using PCM.
MEMs	Micro electrical mechanical Systems. For AVS devices such as Digital MEMs Microphones.
MSI	Message Signaled Interrupt. An in-band method of signaling an interrupt.
PCM	Pulse Code Modulation. Modulation with amplitude coded into stream.
PDM	Pulse Density Modulation. Modulation with amplitude coded by pulse density.
SoC	System On Chip.
VAD	Voice Activity Detector.
VOIP	Voice Over Internet Protocol

12.2 References

High Definition Audio Specification	http://www.intel.com/content/www/us/en/standards/high-definition-audio-specification.html

12.3 Feature Overview

The Platform builds upon the AVS features of previous platforms to provide a richer user experience. This section will cover the HW features used in the PCH for use within the AVS subsystem. The AVS subsystem consists of a collection of controller, DSP, memory, and link interfaces that provides the audio experience to the platform. This subsystem provides streaming of audio from the host SW to external audio codecs, with the host CPU and/or DSP providing the audio enrichment.

The optional DSP can be enabled in the audio subsystem to provide low latency HW/FW acceleration for common audio and voice functions such as audio encode/decode, acoustic echo cancellation, noise cancellation, etc. With such acceleration, the integration of the AVS subsystem into an SOC is expected to provide longer music playback times and VOIP call times for the platform.

The following subsections describe key HW features of the AVS Subsystem.

12.4 Intel® High Definition Audio (Intel® HD Audio) Controller Capabilities

The Intel® HD Audio controller is the standard audio host controller widely adopted in the PC platform, with industrial standard Intel® HD Audio driver software available for Microsoft* Windows* and many other Linux* based OS'es. Intel® HD Audio controller features are listed as follows:

- PCI / PCI Express* controller.
- Supports data transfers, descriptor fetches, and DMA position writes using VC0.
- Independent Bus Master logic for 16 general purpose DMA streams: 7 input and 9 output.
- Supports variable length stream slots.
- Supports up to:
  - 16 streams
  - Seven input
  - Seven system streams
  - Two offload streams
  - Nine output
  - Two system streams (dedicated)
  - Two system / Two offload streams (shared)
  - Two offload streams (dedicated)
  - One feedback stream
  - 16 channels per stream
  - 32 bits/sample
  - 192 kHz sample rate
- Supports memory-based command/response transport.
- Three 8-channel universal DMA interfaces for transferring data between memory buffers and peripherals and between memories
- Supports optional Immediate Command/Response mechanism.
- Supports output and input stream synchronization.
- Supports global time synchronization.
- Supports MSI interrupt delivery.
- Support for ACPI D3 and D0 Device States.
- Supports Function Level Reset (FLR)
- Support Converged Platform Power Management (CPPM).
- Support 1 ms of buffering with all DMA running with maximum bandwidth.
- Support 10 ms of buffering with 1 output DMA and 1 input DMA running at 2 channels, 96 kHz, 16 bit audio.

12.4.1 Audio DSP Capabilities

The Audio DSP offload engine is an optional feature providing low power DSP functionality and offload the audio processing operation from host CPU. It is exposed as an optional capability feature under the Intel® HD Audio controller allowing the enumeration through the Intel® HD Audio driver software (if implemented). Audio DSP features are listed as follows:

- Audio DSP with 4 Tensilica* LX6+HiF3 cores for low power offloaded audio rendering and recording
  - 400 MHz operating frequency in S0
  - 120 MHz operating frequency in S0ix
  - 64KB L1 RAM
  - 3008KB L2 SRAM
- Low power support for Intel® Wake on Voice (Intel® WOV)
- Low power audio playback with post processing
- Low power VoIP and circuit switch voice call with pre-processing
- Various DSP functions provided by DSP Core: MP3, AAC, 3rd Party IP Algorithms, etc.

12.4.2 Intel® High Definition Audio Interface Capabilities

The Intel® HD Audio interface is an optional feature offering connections to the compatible codecs. The Intel® HD Audio compatible codecs are widely available from various vendors allowing PC platform OEM’s to choose them based on features, power, cost consideration. The audio codec can work with the in-box Intel® HD Audio driver software provided in various OS'es providing a seamless user experience. These Intel® HD Audio compatible codecs will be enumerated by the Intel® HD Audio driver software (if discovered over the Intel® HD Audio interface). Intel® HD Audio interface features are listed as follows:

- Two SDI signals to support two external codecs
- Drives variable frequency (6 MHz to 24 MHz) BCLK to support:
  - SDO double pumped up to 48 Mb/s
  - SDIs single pumped up to 24 Mb/s
- Provides cadence for 44.1 kHz-based sample rate output
- Supports 1.5V, 1.8V, and 3.3V modes
  - 1.5V and 1.8V shares the same drive strength programming and 3.3V has its own programming

12.4.3 Direct Attached Digital Microphone (PDM) Interface

The direct attached digital microphone interface is an optional feature offering connections to PDM based digital microphone modules without the need of audio codecs. This provides the lowest possible platform power with the decimation functionality integrated into the audio host controller. Features for the digital microphone interface are listed as follows:

- Two DMIC PDM interfaces with each interface capable of supporting up to 2 digital MEMs microphones
- Low power always listening support for Intel® Wake on Voice (Intel® WOV)

- 2 PCM audio streams (with independent PCM sampling rate: 48 kHz or 16 kHz) per digital mic interface
- Ultrasound reception capable with higher frequency ranges between 3.84 MHz - 4.8 MHz.

12.4.4 I²S/PCM Interface

The I²S / PCM interface is an optional feature offering connection to the I²S / PCM audio codecs. The I²S / PCM audio codecs are widely adopted in the phone and tablet platforms as they are typically customized for low power application. The codec structure is typically unique per codec vendor implementation and requires vendor specific SW module for controlling the codec. These I²S / PCM audio codecs will be enumerated based on ACPI table or OS specific static configuration information. The Audio DSP is required to be enabled in order to enable I²S / PCM link as registers are only addressable through the Audio DSP and its FW. I²S/PCM Interface features are listed as follows:

- Multiple I²S/PCM ports to support multiple I²S connections
- Can support 3 modes: Slave Mode, Slave Mode with Locally Generated Master Clock, or Master Mode
- I²S audio playback at 2 ch x 192 kHz x 24 bits
- I²S audio capture at 2 ch x 192 kHz x 24 bits
- PCM audio playback at 8 ch x 48 kHz x 24 bits
- PCM audio capture at 8 ch x 48 kHz x 24 bits
- Support 3G / 4G modem codec
- Support BT codec HFP / HSP SCO at 8 / 16 kHz
- Support BT codec A2DP at 48 kHz
- Support FM radio codec

12.4.5 SoundWire* Interface

The SoundWire* interface is an optional feature offering connection to the SoundWire* devices, which include audio codecs and modem codecs. The SoundWire* interface is the latest audio interface targeting (but not limited to) the phone and tablet market and the main advantage is the connection simplicity with a 2 wires multi-drop topology + PCM/PDM streaming capabilities. Currently SoundWire* devices are non-standard across different vendors (similar to I²S / PCM audio codecs), hence it is very likely to require customized audio codec SW per vendor. These devices will be enumerated based on vendor / device ID of the SoundWire device reporting. SoundWire* interface features are listed as follows:

- 4 independent SoundWire* Interfaces for connection to audio peripherals
- Single audio playback at 8 ch x 96 kHz x 24 bits
- Up to 2 concurrent audio playback at 2 ch x 192 kHz x 24 bit each
- Single audio capture at 8 ch x 96 kHz x 24 bits
- Up to 4 concurrent audio capture of 2 ch x 96 kHz x 24 bit each
- Up to 16 PCM bidirectional streams per SoundWire* interface
- Direction is programmable as either input or output stream

- Up to 8 PDM input streams per SoundWire* interface
- Up to 8 PDM output streams per SoundWire* interface
- Up to 2 channels per PCM streams
- Up to 1 channel per PDM streams
- Ability to map each stereo PCM streams to a sub-set of a multi-channel PCM stream DMA data transferred over Audio Link Hub
- Ability to map each mono PDM input stream to a sub-set of a multi-channel PDM stream DMA data transferred to digital mic port (decimation input)
- Ability to map each mono PDM output stream to a sub-set of a multi-channel PDM stream DMA data transferred from digital speaker port (interpolation output)

12.5 Signal Description

Table 12-1. Signal Descriptions (Sheet 1 of 3)

Name	Type	Description
Intel®High Definition Audio Signals		
HDA_RST#	O	**Intel® HD Audio Reset**: Master H/W reset to internal/external codecs.
HDA_SYNC	O	**Intel® HD Audio Sync:** 48 kHz fixed rate frame sync to the codecs. Also used to encode the stream number.
HDA_BCLK	O	**Intel® HD Audio Bit Clock:** Up to 24 MHz serial data clock generated by the Intel® HD Audio controller.
HDA_SDO	O	**Intel® HD Audio Serial Data Out:** Serial TDM data output to the codecs. The serial output is double-pumped for a bit rate of up to 48 Mb/s.
HDA_SDI0	I	**Intel® HD Audio Serial Data In 0:** Serial TDM data input from the two codec(s). The serial input is single-pumped for a bit rate of up to 24 Mb/s. These signals contain integrated Pull-down resistors, which are enabled while the primary well is powered.
HDA_SDI1/ I2S1_SFRM/ GPP_R5	I	**Intel® HD Audio Serial Data In 1:** Serial TDM data input from the two codec(s). The serial input is single-pumped for a bit rate of up to 24 Mb/s. These signals contain integrated Pull-down resistors, which are enabled while the primary well is powered.
I²S/PCM Interface		
I2S0_SCLK	I/O	**I²S/PCM serial bit clock 0:** Clock used to control the timing of a transfer. Can be generated internally (Master mode) or taken from an external source (Slave mode).
I2S3_SCLK	I/O	**I²S/PCM serial bit clock 3:** Clock used to control the timing of a transfer. Can be generated internally (Master mode) or taken from an external source (Slave mode).
I2S4_SCLK/ GPPC_A15/ USB_OC2#/ DDSP_HPD4/ DISP_MISC4	I/O	**I²S/PCM serial bit clock 4:** This clock is used to control the timing of a transfer. Can be generated internally (Master mode) or taken from an external source (Slave mode).
I2S5_SCLK/ GPPC_A19/ DDSP_HPD1/ DISP_MISC1	I/O	**I²S/PCM serial bit clock 5:** This clock is used to control the timing of a transfer. Can be generated internally (Master mode) or taken from an external source (Slave mode).
I2S0_SFRM/ HDA_SYNC / GPP_R1	I/O	**I²S/PCM serial frame indicator 0**: This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Master mode) or taken from an external source (Slave mode).
I2S1_SFRM/ HDA_SDI_1/ GPP_R5	I/O	**I²S/PCM serial frame indicator 1**: This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Master mode) or taken from an external source (Slave mode).

Table 12-1. Signal Descriptions (Sheet 2 of 3)

Name	Type	Description
I2S2_SFRM/ GPPC_A8	I/O	**I²S/PCM serial frame indicator 2**: This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Master mode) or taken from an external source (Slave mode).
I2S0_TXD	O	**I²S/PCM transmit data (serial data out)0**: This signal transmits serialized data. The sample length is a function of the selected serial data sample size.
I2S1_TXD/ GPP_R6	O	**I²S/PCM transmit data (serial data out)1**: This signal transmits serialized data. The sample length is a function of the selected serial data sample size.
I2S2_TXD/ GPPC_A9	O	**I²S/PCM transmit data (serial data out)2**: This signal transmits serialized data. The sample length is a function of the selected serial data sample size.
I2S0_RXD/ HDA_SDI0/ GPP_R3	I	**I²S/PCM receive data (serial data in)0:** This signal receives serialized data. The sample length is a function of the selected serial data sample size.
I2S1_RXD/ GPP_R7	I	**I²S/PCM receive data (serial data in)1:** This signal receives serialized data. The sample length is a function of the selected serial data sample size.
I2S2_RXD/ GPPC_A10	I	**I²S/PCM receive data (serial data in)2:** This signal receives serialized data. The sample length is a function of the selected serial data sample size.
I2S_MCLK/ GPPC_D19	O	**I²S/PCM Master reference clock:** This signal is the master reference clock that connects to an audio codec.
DMIC Interface		
DMIC_CLK0/ GPP_S6/ SNDW4_CLK	O	**Digital Mic Clock:** Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz.
DMIC_CLK1/ GPP_S4/ SNDW3_CLK	O	**Digital Mic Clock:** Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz.
DMIC_DATA0 /GPP_S7/ SNDW4_DATA	I	**Digital Mic Data:** Serial data input from the digital mic.
DMIC_DATA1 /GPP_S5/ SNDW3_DATA	I	**Digital Mic Data:** Serial data input from the digital mic.
SoundWire* Interface		
SNDW1_CLK/ GPP_S0	I/O	**SoundWire* Clock:** Serial data clock to external peripheral devices.
SNDW2 _CLK / GPP_S2	I/O	**SoundWire* Clock:** Serial data clock to external peripheral devices.
SNDW3_CLK/ GPPC_S4/ DMIC_CLK1	I/O	**SoundWire* Clock:** Serial data clock to external peripheral devices.
SNDW4_CLK/ GPPC_S6/ DMIC_CLK0	I/O	**SoundWire* Clock:** Serial data clock to external peripheral devices.
SNDW1_DAT A/GPP_S1	I/O	**SoundWire* Data:** Serial data input from external peripheral devices.
SNDW2 _DATA / GPP_S3	I/O	**SoundWire* Data:** Serial data input from external peripheral devices.
SNDW3_DAT A/ GPP_S5/ DMIC_DATA1	I/O	**SoundWire*Data:** Serial data input from external peripheral devices.
SNDW4_DAT A/GPP_S7/ DMIC_DATA0	I/O	**SoundWire* Data:** Serial data input from external peripheral devices.
Misc		

Table 12-1. Signal Descriptions (Sheet 3 of 3)

Name	Type	Description
SPKR/ GPPC_B14	O	**Speaker Output:** Used for connection to external speaker for POST sounds if not using HD_Audio embedded option.

12.6 Integrated Pull-Ups and Pull-Downs

Table 12-2. Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value (Ω)	Notes
HDA_SYNC	Pull-down	14K-26K	
HDA_SDO	Pull-down	14K-26K	
HDA_SDI[1:0]	Pull-down	14K-26K	
I2S0_TXD	Pull-down	14K-26K	
SNDW_DATA[4:1]	Pull-down	14K-26K	

12.7 I/O Signal Planes and States

Table 12-3. I/O Signal Planes and States (Sheet 1 of 2)

Signal Name	Power Plane	During Reset[2]	Immediately After Reset[2]	S3/S4/S5	Deep Sx
High Definition Audio Interface					
HDA_RST#	Primary	Driven Low	Driven Low	Driven Low	OFF
HDA_SYNC	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
HDA_BLK	Primary	Driven Low	Driven Low	Driven Low	OFF
HDA_SDO	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
HDA_SDI[1:0]	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I²S/PCM Interface					
I2S[2:1]_SCLK	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S[2:0]_SFRM	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S0_TXD	Primary	Internal Pull-down	Driven Low	Low then disabled (Refer Note)	OFF
I2S[2:1]_TXD	Primary	Driven Low	Driven Low	Driven Low	OFF
I2S[2:0]_RXD	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S_MCLK	Primary	Driven Low	Driven Low	Driven Low	OFF
DMIC Interface					
DMIC_CLK[1:0]	Primary	Driven Low	Driven Low	Driven Low	OFF
DMIC_DATA[1:0]	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SoundWire* Interface					

Table 12-3. I/O Signal Planes and States (Sheet 2 of 2)

Signal Name	Power Plane	During Reset[2]	Immediately After Reset[2]	S3/S4/S5	Deep Sx
SNDW_DATA[1:4]	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SNDW_CLK[1:4]	Primary	Driven Low	Driven Low	Driven Low	OFF
Misc					
SPKR	Primary	Internal Pull-down	Driven Low	Low then disabled (Refer note)	OFF

Notes:
1. SPKR and I2S0_TXD are also straps in which the pull-down only occurs during the sampling window and then the pull-ups are disabled.
2. Reset reference for primary well pins is RSMRST#.

§ §

13 Controller Link

13.1 Acronyms

Acronyms	Description
CL	Controller Link
WLAN	Wireless Local Area Network

13.2 Feature Overview

The controller link is used to manage the wireless devices supporting Intel® Active Management Technology. Controller Link will transmit data at 60.0 Mbps on Controller Link Port. The Controller Link clock frequency is 30.0 MHz, MCLK will operate at 30.0 MHz.

13.3 Signal Description

Name	Type	Description
CL_DATA	I/O	**Controller Link Data:** Bi-directional data that connects to a Wireless LAN Device supporting Intel® Active Management Technology.
CL_CLK	I/O	**Controller Link Clock**: Bi-directional clock that connects to a Wireless LAN Device supporting Intel® Active Management Technology.
CL_RST#	O	**Controller Link Reset:** Controller Link reset that connects to a Wireless LAN Device supporting Intel® Active Management Technology.

13.4 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value (Ohm)	Notes
CL_DATA	Pull-up Pull-down	31.25 100	Refer Section 13.5
CL_CLK	Pull-up Pull-down	31.25 100	Refer Section 13.5

13.5 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[3]	Immediately after Reset[3]	S3/S4/S5	Deep Sx
CL_DATA	Primary	Refer Notes	Refer Notes	Internal Pull-down	Off
CL_CLK	Primary	Refer Notes	Refer Notes	Internal Pull-down	Off
CL_RST#	Primary	Driven Low	Driven High	Driven High	Off

Notes:
1. The Controller Link clock and data buffers use internal Pull-up or Pull-down resistors to drive a logical 1 or 0.
2. The terminated state is when the I/O buffer Pull-down is enabled.
3. Reset reference for primary well pins is RSMRST#.

13.6 External CL_RST# Pin Driven/Open-drained Mode Support

The WLAN has transitioned to 1.8V for external CL_RST# pin, while PCH Controller Link I/O buffer still drives 3.3V on this pin. This creates voltage in-compatibility issue. In order to support either 1.8V or 3.3V on the device CL_RST# pin, the PCH operates/ controls the CL_RST# pin as dual modes, which is determined by a Soft-strap bit:

(1) Driven mode: To drive "1" on this pin, Controller Link turn-on the output enable and output=1 to drive 3.3V on this pin. This mode can only be enabled with older version of WLAN which is 3.3V tolerant.

(2) Open-drained mode: To drive "1", Controller Link turn-off the output-enable, and external (required) pull-up will pull the pin up to 1.8V, which is compatible with WLAN voltage requirement.

§ §

14 Processor Sideband Signals

14.1 Acronyms

Acronyms	Description
PECI	Platform Environmental Control Interface

14.2 Feature Overview

The sideband signals are used for the communication between the processor and PCH.

14.3 Signal Description

Name	Type	Description
PROCPWRGD	O	Signal to the processor to indicate its primary power is good.
THRMTRIP#	I	Signal from the processor to indicate that a thermal overheating has occurred.
PECI	I/O	Single-wire serial bus for accessing processor digital thermometer
CPU_GP0/GPP_E3	I	Thermal management signal
CPU_GP1/GPP_E7	I	Thermal management signal
CPU_GP2/GPP_B3	I	Thermal management signal
CPU_GP3/GPP_B4	I	Thermal management signal

14.4 Integrated Pull-Ups and Pull-Downs

None

14.5 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S3/S4/S5	Deep Sx
PROCPWRGD	Primary	Undriven	Driven High	Off	Off
THRMTRIP#	Primary	Undriven	Undriven	Off	Off
PECI	Primary	Undriven	Undriven	Off	Off
CPU_GP[3:0]	Primary	Undriven	Undriven	Undriven	Off

Notes:
1. Reset reference for primary well pins is RSMRST#.

14.6 Functional Description

PROCPWRGD out to the processor indicates that the primary power is ramped up and stable. PROCPWRGD will be undriven by the PCH (high Z) when RSMRST# is asserted and driven high after RSMRST# is de-asserted.

If THRMTRIP# goes active, the processor is indicating an overheat condition, and the PCH will immediately transition to an S5 state. CPU_GP can be used from external sensors for the thermal management.

PM_SYNC is used to provide early warning to the processor that a global reset is in progress and that the memory contents should be saved and placed into self refresh.

PM_DOWN is input to PCH indicates the processor wake up event.

§ §

15 Digital Display Signals

15.1 Acronyms

Acronyms	Description
eDP*	embedded Display Port*

15.2 Signal Description

Display is divided between processor and PCH. The processor houses memory interface, display planes, pipes, and digital display interfaces/ports while the PCH has DDC bus, Hot-plug Detect, Panel Power and Back light controls.

The DDC (Digital Display Channel) bus is used for communication between the host system and display. Two pairs of DDC (DDC_CLK and DDC_DATA) signals exist on the PCH that correspond to two digital ports on the processor. DDC is based on I^2C protocol.

The Hot-Plug Detect (HPD) signal serves as an interrupt request for the sink device for DisplayPort* and HDMI*. It is a 3.3V tolerant signal pin on the PCH.

The Panel Power and Back light controls are used to control power for an internal panel and drive the back light

Table 15-1. Digital Display Signals (Sheet 1 of 2)

Name	Type	Description
DPPE_HPDA / GPP_E14 / DISP_MISCA	I	**Display Port A:** HPD Hot-Plug Detect
DDSP_HPDB /GPP_A18 / DISP_MISCB	I	**Display Port B:** HPD Hot-Plug Detect
DDSP_HPD1 /GPP_A19 / DISP_MISC1	I	**Display Port C:** HPD Hot-Plug Detect
DDSP_HPD2 / GPP_A20 / DISP_MISC2	I	**Display Port D:** HPD Hot-Plug Detect
DDSP_HPD3 /GPP_A14 / USB_OC1# / DISP_MISC3	I	**Display Port E:** HPD Hot-Plug Detect
DDSP_HPD4 /GPP_A15 / USB_OC2# / DISP_MISC4	I	**Display Port F:** HPD Hot-Plug Detect
DDPA_CTRLCLK /GPP_E22 / PCIE_LNK_DOWN	I/O	**Display Port A:** Control Clock.
DDPA_CTRLDATA /GPP_E23 / BK4 / SBK4	I/O	**Display Port A:** Control Data.
DDPB_CTRLCLK / GPP_H16	I/O	**Display Port B:** Control Clock.
DDPB_CTRLDATA / GPP_H17	I/O	**Display Port B:** Control Data.
DDP1_CTRLCLK /GPP_E18 / TBT_LSX0_TXD	I/O	**Display Port C:** Control Clock.
DDP1_CTRLDATA /GPP_E19 / TBT_LSX0_RXD	I/O	**Display Port C:** Control Data.
DDP2_CTRLCLK /GPP_E20 / TBT_LSX1_TXD	I/O	**Display Port D:** Control Clock.
DDP2_CTRLDATA /GPP_E21 / TBT_LSX1_RXD	I/O	**Display Port D:** Control Data.
DDP3_CTRLCLK /GPP_D9 / ISH_SPI_CS# / GSPI2_CS0# / TBT_LSX2_TXD	I/O	**Display Port E:** Control Clock.
DDP3_CTRLDATA /GPP_D10 / ISH_SPI_CLK / GSPI2_CLK / TBT_LSX2_RXD	I/O	**Display Port E:** Control Data.

Table 15-1. Digital Display Signals (Sheet 2 of 2)

Name	Type	Description
DDP4_CTRLCLK /GPP_D11 / ISH_SPI_MISO / GSPI2_MISO / TBT_LSX3_TXD	I/O	**Display Port F:** Control Clock.
DDP4_CTRLDATA /GPP_D12 / ISH_SPI_MOSI / GSPI2_MOSI / TBT_LSX3_RXD	I/O	**Display Port F:** Control Data.

15.3 Embedded DisplayPort* (eDP*) Backlight Control Signals

eDP_VDDEN	O	**eDP* Panel Power Enable:** Panel power control enable. This signal is used to control the VDC source of the panel logic.
eDP_BKLTEN	O	**eDP* Backlight Enable:** Panel backlight enable control for eDP* This signal is used to gate power into the backlight circuitry.
eDP_BKLTCTL	O	**eDP* Panel Backlight Brightness control:** Panel brightness control for eDP*. This signal is used as the PWM Clock input signal
Note: eDP_VDDEN, eDP_BKLTEN, eDP_BKLTCTL can be left as no connect if eDP* is not used.		

15.4 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
DDPA_CTRLDATA	Pull-down	15K-40K	Refer note below
DDPB_CTRLDATA	Pull-down	15K-40K	Refer note below
DDP1_CTRLDATA	Pull-down	15K-40K	Refer note below
DDP2_CTRLDATA	Pull-down	15K-40K	Refer note below
DDP3_CTRLDATA	Pull-down	15K-40K	Refer note below
DDP4_CTRLDATA	Pull-down	15K-40K	Refer note below

Note: The internal pull-up/pull-down is only applied during the strap sampling window (PCH_PWROK) and is then disabled. Enabling can be done using a 2.2 KOhm Pull-up resistor.

15.5 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S3/S4/S5	Deep Sx
DPPE_HPDA	Primary	Undriven	Undriven	Undriven	OFF
DDSP_HPDB	Primary	Undriven	Undriven	Undriven	OFF
DDSP_HPD1	Primary	Undriven	Undriven	Undriven	OFF
DDSP_HPD2	Primary	Undriven	Undriven	Undriven	OFF
DDSP_HPD3	Primary	Undriven	Undriven	Undriven	OFF
DDSP_HPD4	Primary	Undriven	Undriven	Undriven	OFF
DDPA_CTRLCLK	Primary	Undriven	Undriven	Undriven	OFF
DDPA_CTRLDATA	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
DDPB_CTRLCLK	Primary	Undriven	Undriven	Undriven	OFF

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S3/S4/S5	Deep Sx
DDPB_CTRLDATA	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
DDP1_CTRLCLK	Primary	Undriven	Undriven	Undriven	OFF
DDP1_CTRLDATA	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
DDP2_CTRLCLK	Primary	Undriven	Undriven	Undriven	OFF
DDP2_CTRLDATA	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
DDP3_CTRLCLK	Primary	Undriven	Undriven	Undriven	OFF
DDP3_CTRLDATA	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
DDP4_CTRLCLK	Primary	Undriven	Undriven	Undriven	OFF
DDP4_CTRLDATA	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
eDP_VDDEN	Primary	Driven Low	Driven Low	Driven Low	OFF
eDP_BKLTEN	Primary	Driven Low	Driven Low	Driven Low	OFF
eDP_BKLTCTL	Primary	Driven Low	Driven Low	Driven Low	OFF

Notes:
1. Reset reference for primary well pins is RSMRST#.

§ §

16 Enhanced Serial Peripheral Interface (eSPI)

16.1 Acronyms

Acronyms	Description
EC	Embedded Controller
MAFCC	Master Attached Flash Channel Controller (MAFCC)
OOB	Out-of-Band
TAR	Turn-around cycle

16.2 Feature Overview

The PCH provides the Enhanced Serial Peripheral Interface (eSPI) to support connection of an EC (typically used in mobile platform) or an SIO (typically used in desktop platform) to the platform. Below are the key features of the interface:

- 1.8V support only
- Support for Master Attached Flash and Slave Attached Flash.
- Support for 20 MHz, 24 MHz, 30 MHz, 48 MHz, and 60 MHz (configured by soft straps)
- Up to quad mode support
- Support for PECI over eSPI
- Support for Multiple OOB Master (dedicated OOB channel for different OOB masters in the PCH such as PMC and ME)
- Transmitting RTC time/date to the slave device upon request
- In-band messages for communication between the PCH and slave device to eliminate side-band signals
- Real time SPI flash sharing, allowing real time operational access by the PCH and slave device

16.3 Signal Description

Name	Type	Description
ESPI_IO0/GPP_A0	I/O	**eSPI Data Signal 0:** Bi-directional pin used to transfer data between the PCH and eSPI slave device.
ESPI_IO1/GPP_A1	I/O	**eSPI Data Signal 1:** Bi-directional pin used to transfer data between the PCH and eSPI slave device
ESPI_IO2/GPP_A2	I/O	**eSPI Data Signal 2:** Bi-directional pin used to transfer data between the PCH and eSPI slave device
ESPI_IO3/GPP_A3	I/O	**eSPI Data Signal 3:** Bi-directional pin used to transfer data between the PCH and eSPI slave device

Name	Type	Description
ESPI_CS#/GPP_A4	O	**eSPI Chip Select:** Driving CS# signal low to select eSPI slave for the transaction.
ESPI_CLK/GPP_A5	O	**eSPI Clock:** eSPI clock output from the PCH to slave device.
ESPI_RESET#/ GPP_A6	O	**eSPI Reset:** Reset signal from the PCH to eSPI slave.

16.4 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
ESPI_IO[3:0]	Pull-up	20K +/- 30%	
ESPI_CLK	Pull-down	20K +/- 30%	
ESPI_ CS#	Pull-up	20K +/- 30%	

16.5 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S3/S4/S5	Deep Sx
ESPI_IO [3:0]	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up	OFF
ESPI_CLK	Primary	Internal Pull-down	Driven Low	Driven Low	OFF
ESPI_ CS#	Primary	Internal Pull-up	Driven High	Driven High	OFF
ESPI_RESET#	Primary	Driven Low	Driven High	Driven High	OFF

Notes:
1. Reset reference for primary well pins is RSMRST#.

16.6 Functional Description

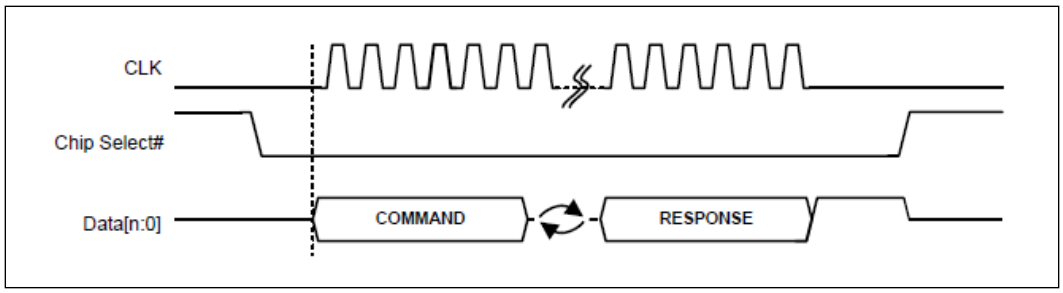
16.6.1 Operating Frequency

The eSPI controller supports 20 MHz, 24 MHz, 30 MHz, 48 MHz, and 60 MHz. A slave device can support frequencies lower than the recommended maximum frequency (60 MHz). In addition, the slave device must support a minimum frequency of 20 MHz for default (reset) communication between the Master and Slave device.

16.6.2 Protocols

Below is an overview of the basic eSPI protocol. Refer to the latest eSPI Specification and corresponding platform eSPI Compatibility Specification for more details (Refer Section 16).

Figure 16-1. Basic eSPI Protocol



An eSPI transaction consists of a Command phase driven by the master, a turn-around phase (TAR), and a Response phase driven by the slave.

A transaction is initiated by the PCH through the assertion of CS#, starting the clock and driving the command onto the data bus. The clock remains toggling until the complete response phase has been received from the slave.

The serial clock must be low at the assertion edge of the CS# while ESPI_RESET# has been de-asserted. The first data is driven out from the PCH while the serial clock is still low and sampled on the rising edge of the clock by the slave. Subsequent data is driven on the falling edge of the clock from the PCH and sampled on the rising edge of the clock by the slave. Data from the slave is driven out on the falling edge of the clock and is sampled on a falling edge of the clock by the PCH.

All transactions on eSPI are in multiple of 8 bits (one byte).

16.6.3 WAIT States from eSPI Slave

There are situations when the slave cannot predict the length of the command packet from the master (PCH). For non-posted transactions, the slave is allowed to respond with a limited number of WAIT states.

A WAIT state is a 1-byte response code. They must be the first set of response byte from the slave after the TAR cycles.

16.6.4 In-Band Link Reset

In case the eSPI link may end up in an undefined state (for example when a CRC error is received from the slave in a response to a Set_Configuration command), the PCH issues an In-Band Reset command that resets the eSPI link to the default configuration. This allows the controller to re-initialize the link and reconfigure the slave.

16.6.5 Slave Discovery

The PCH eSPI interface is enabled using a hard pin strap. If this strap is asserted (high) at RSMRST# de-assertion, the eSPI controller is enabled and assumes that a slave is connected to the interface. The controller does not perform any other discovery to confirm the presence of the slave connection.

If the ESPI_EN HW strap is de-asserted (low), the eSPI controller will gate all its clocks and put itself to sleep.

16.6.6 Flash Sharing Mode

eSPI supports both Master and Slave Attached Flash sharing (abbreviated in this as MAFS and SAFS, respectively). The Flash sharing mode selected for a specific platform is dependent on strap settings.

In order for SAFS to work, the Slave must support the Flash Access channel.

16.6.7 PECI Over eSPI

When PECI Over eSPI is enabled, the eSPI device (i.e. EC) can access the processor PECI interface via eSPI controller, instead of the physical PECI pin. The support can improve the PECI responsiveness, and reduce PECI pins.

The PECI bus may be connected to the PCH via either the legacy PECI pin or the eSPI interface. The operation via legacy PECI pin or over eSPI is selected via a soft strap and only one or the other is enabled in a given platform.

PECI over eSPI is not supported in Sx state. EC/BMC is not allowed to send the PECI command to eSPI in Sx states. More specifically, EC can only send PECI requests after VW PLT_RST# de-assertion.

In S0ix, upon receiving a PECI command, the PMC will wake up the CPU from Cx and respond back once the data is available from CPU.

16.6.8 Multiple OOB Master

PCHs typically have multiple embedded processors (ME, PMC, ISH, etc.). From an eSPI perspective, these are all classified as Out-of-Band (OOB) processors (as distinct from the Host processor). Since any of these OOB processors may need to communicate with the embedded controller on the platform (example, EC. BMC), the eSPI controller implements dedicated OOB channel for each OOB processors including PMC and ME to improve the interface performance and potentially enable new usage models.

16.6.9 Channels and Supported Transactions

An eSPI channel provides a means to allow multiple independent flows of traffic to share the same physical bus. Refer to the eSPI specification for more detail.

Each of the channels has its dedicated resources such as queue and flow control. There is no ordering requirement between traffic from different channels.

The number of types of channels supported by a particular eSPI slave is discovered through the GET_CONFIGURATION command issued by the PCH to the eSPI slave during initialization.

Table 16-1 summarizes the eSPI channels and supported transactions.

Table 16-1. eSPI Channels and Supported Transactions (Sheet 1 of 2)

CH #	Channel	Posted Cycles Supported	Non-Posted Cycles Supported
0	Peripheral	Memory Write, Completions	Memory Read, I/O Read/Write
1	Virtual Wire	Virtual Wire GET/PUT	N/A
2	Out-of-Band Message	SMBus Packet GET/PUT	N/A

Table 16-1. eSPI Channels and Supported Transactions (Sheet 2 of 2)

CH #	Channel	Posted Cycles Supported	Non-Posted Cycles Supported
3	Flash Access	N/A	Flash Read, Write, Erase
N/A	General	Register Accesses	N/A

16.6.9.1 Peripheral Channel (Channel 0) Overview

The Peripheral channel performs the following Functions:

- Target for PCI Device D31:F0: The eSPI controller duplicates the legacy LPC PCI Configuration space registers. These registers are mostly accessed via the BIOS, though some are accessed via the OS as well.
- Tunnel all Host to eSPI slave (EC/SIO) debug device accesses: these are the accesses that used to go over the LPC bus. These include various programmable and fixed I/O ranges as well as programmable Memory ranges. The programmable ranges and their enables reside in the PCI Configuration space.
- Tunnel all accesses from the eSPI slave to the Host. These include Memory Reads and Writes.

16.6.9.2 Virtual Wire Channel (Channel 1) Overview

The Virtual Wire channel uses a standard message format to communicate several types of signals between the components on the platform.

- Sideband and GPIO Pins: System events and other dedicated signals between the PCH and eSPI slave. These signals are tunneled between the 2 components over eSPI.
- Serial IRQ Interrupts: Interrupts are tunneled from the eSPI slave to the PCH. Both edge and triggered interrupts are supported.

16.6.9.2.1 eSPI Virtual Wires (VW)

Table 16-2 summarizes the PCH virtual wires in eSPI mode.

Table 16-2. eSPI Virtual Wires (VW) (Sheet 1 of 2)

Virtual Wire	PCH Pin Direction	Reset Control	Pin Retained in PCH (For Use by Other Components)
SUS_STAT#	Output	ESPI_RESET#	No
SUSWARN#	Output	ESPI_RESET#	No
SUS_ACK	Input	ESPI_RESET#	No
SUSPWRDNACK	Output	ESPI_RESET#	No
PLTRST#	Output	ESPI_RESET#	Yes
PME# (eSPI Peripheral PME)	Input	ESPI_RESET#	N/A
WAKE#	Input	ESPI_RESET#	No
SMI#	Input	PLTRST#	N/A
SCI#	Input	PLTRST#	N/A
RCIN#	Input	PLTRST#	No
SLP_A#	Output	ESPI_RESET#	Yes

Table 16-2. eSPI Virtual Wires (VW) (Sheet 2 of 2)

Virtual Wire	PCH Pin Direction	Reset Control	Pin Retained in PCH (For Use by Other Components)
SLP_S3#/SLP_S4#/ SLP_S5#/SLP_LAN#/ SLP_WLAN#	Output	DSW_PWROK	Yes
SLAVE_BOOT_LOAD_DONE	Input	ESPI_RESET#	N/A
SLAVE_BOOT_LOAD_STATUS	Input	ESPI_RESET#	N/A
HOST_RST_WARN	Output	PLTRST#	N/A
HOST_RST_ACK	Input	PLTRST#	N/A
OOB_RST_WARN	Output	ESPI_RESET#	N/A
OOB_RST_ACK	Input	ESPI_RESET#	N/A
HOST_C10	Output	PLTRST#	N/A
ERROR_NONFATAL	Input	ESPI_RESET#	N/A
ERROR_FATAL	Input	ESPI_RESET#	N/A

16.6.9.2.2 Interrupt Events

eSPI supports both level and edge-triggered interrupts. Refer to the eSPI Specification for details on the theory of operation for interrupts over eSPI.

The PCH eSPI controller will issue a message to the PCH interrupt controller when it receives an IRQ group in its VW packet, indicating a state change for that IRQ line number.

The eSPI slave can send multiple VW IRQ index groups in a single eSPI packet, up to the Operating Maximum VW Count programmed in its Virtual Wire Capabilities and Configuration Channel.

The eSPI controller acts only as a transport for all interrupt events generated from the slave. It does not maintain interrupt state, polarity or enable for any of the interrupt events.

16.6.9.3 Out-of-Band Channel (Channel 2) Overview

The Out-of-Band channel performs the following Functions:

- Tunnel MCTP Packets between the Intel® CSME and eSPI slave device: The Intel® CSME communicates MCTP messages to/from the device by embedding those packets over the eSPI protocol. This eliminates the SMBus connection between the PCH and the slave device which was used to communicate the MCTP messages in prior PCH generations. The eSPI controller simply acts as a message transport and forwards the packets between the Intel® CSME and eSPI device.
- Tunnel PCH Temperature Data to the eSPI slave: The eSPI controller stores the PCH temperature data internally and sends it to the slave using a posted OOB message when a request is made to a specific destination address.
- Tunnel PCH RTC Time and Date Bytes to the eSPI slave: the eSPI controller captures this data internally at periodic intervals from the PCH RTC controller and sends it to the slave device using a posted OOB message when a request is made to a specific destination address.

16.6.9.3.1 PCH Temperature Data Over eSPI OOB Channel

eSPI controller supports the transmitting of PCH thermal data to the eSPI slave. The thermal data consists of 1 byte of PCH temperature data that is transmitted periodically (~1 ms) from the thermal sensor unit.

The packet formats for the temperature request from the eSPI slave and the PCH response back are shown in Figure 16-2 and Figure 16-3.

Figure 16-2. eSPI Slave Request to PCH for PCH Temperature

eSPI Slave to PCH: Request for PCH Temperature

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message = 21h							
1	Tag[3:0]				Length[11:8] = 0h			
2	Length[7:0] = 04h							
3	Destination Slave Addr. = 02h (PCH OOB HW Handler)							0
4	Command Code = 01h (Get_PCH_Temp)							
5	Byte Count = 01h							
6	Source Slave Address = 0Fh (eSPI Slave 0 [EC])							1

Figure 16-3. PCH Response to eSPI Slave with PCH Temperature

PCH to eSPI Slave: Response with PCH Temperature

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message = 21h							
1	Tag[3:0]				Length[11:8] = 0h			
2	Length[7:0] = 05h							
3	Destination Slave Addr. = 0Fh (eSPI Slave 0 [EC])							0
4	Command Code = 01h (Get_PCH_Temp)							
5	Byte Count = 02h							
6	Source Slave Addr. = 02h (PCH OOB HW Handler)							1
7	PCH Temperature Data [7:0]							

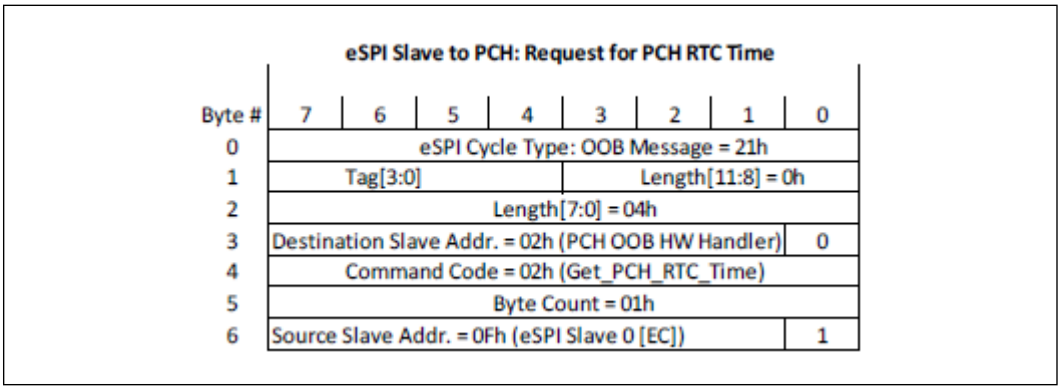
16.6.9.3.2 PCH RTC Time/Date to EC Over eSPI OOB Channel

The PCH eSPI controller supports the transmitting of PCH RTC time/date to the eSPI slave. This allows the eSPI slave to synchronize with the PCH RTC system time. Moreover, using the OOB message channel allows reading of the internal time when the system is in Sx states.

The RTC time consists of 7 bytes: seconds, minutes, hours, day of week, day of month, month and year. The controller provides all the time/date bytes together in a single OOB message packet. This avoids the boundary condition of possible roll over on the RTC time bytes if each of the hours, minutes, and seconds bytes is read separately.

The packet formats for the RTC time/date request from the eSPI slave and the PCH response back to the device are shown in Figure 16-4 and Figure 16-5.

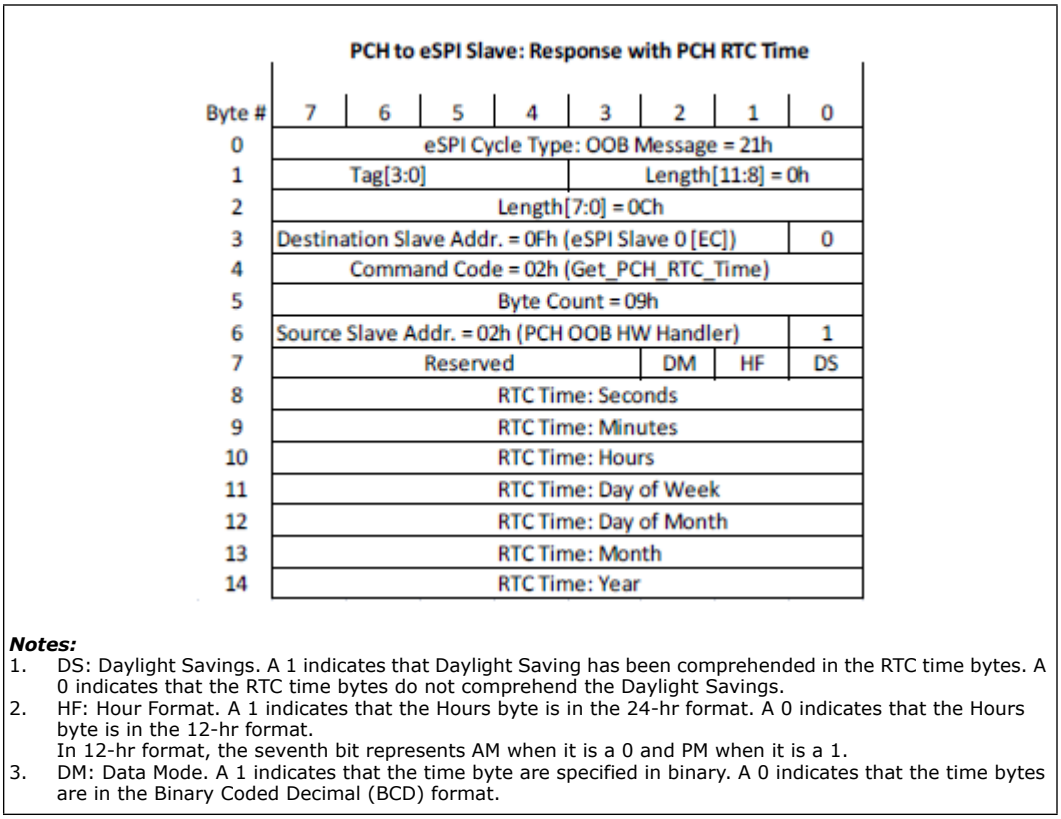
Figure 16-4. eSPI Slave Request to PCH for PCH RTC Time



eSPI Slave to PCH: Request for PCH RTC Time

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message = 21h							
1	Tag[3:0]				Length[11:8] = 0h			
2	Length[7:0] = 04h							
3	Destination Slave Addr. = 02h (PCH OOB HW Handler)							0
4	Command Code = 02h (Get_PCH_RTC_Time)							
5	Byte Count = 01h							
6	Source Slave Addr. = 0Fh (eSPI Slave 0 [EC])							1

Figure 16-5. PCH Response to eSPI Slave with RTC Time



PCH to eSPI Slave: Response with PCH RTC Time

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message = 21h							
1	Tag[3:0]				Length[11:8] = 0h			
2	Length[7:0] = 0Ch							
3	Destination Slave Addr. = 0Fh (eSPI Slave 0 [EC])							0
4	Command Code = 02h (Get_PCH_RTC_Time)							
5	Byte Count = 09h							
6	Source Slave Addr. = 02h (PCH OOB HW Handler)							1
7	Reserved					DM	HF	DS
8	RTC Time: Seconds							
9	RTC Time: Minutes							
10	RTC Time: Hours							
11	RTC Time: Day of Week							
12	RTC Time: Day of Month							
13	RTC Time: Month							
14	RTC Time: Year							

Notes:
1. DS: Daylight Savings. A 1 indicates that Daylight Saving has been comprehended in the RTC time bytes. A 0 indicates that the RTC time bytes do not comprehend the Daylight Savings.
2. HF: Hour Format. A 1 indicates that the Hours byte is in the 24-hr format. A 0 indicates that the Hours byte is in the 12-hr format.
   In 12-hr format, the seventh bit represents AM when it is a 0 and PM when it is a 1.
3. DM: Data Mode. A 1 indicates that the time byte are specified in binary. A 0 indicates that the time bytes are in the Binary Coded Decimal (BCD) format.

16.6.9.4 Flash Access Channel (Channel 3) Overview

The Flash Access channel supports the Master Attached Flash (MAF) configuration, where the flash device is directly attached to the PCH. This configuration allows the eSPI device to access the flash device attached to the PCH through a set of flash access commands. These commands are routed to the flash controller and the return data is sent back to the eSPI device.

The Master Attached Flash Channel controller (MAFCC) tunnels flash accesses from eSPI slave to the PCH flash controller. The MAFCC simply provides Flash Cycle Type, Address, Length, Payload (for writes) to the flash controller. The flash controller is responsible for all the low level flash operations to perform the requested command and provides a return data/status back to the MAFCC, which then tunnels it back to the eSPI slave in a separate completion packet.

16.6.9.4.1 Master Attached Flash Channel Controller (MAFCC) Flash Operations and Addressing

The EC is allocated a dedicated region within the eSPI Master-Attached flash device. The EC has default read, write, and erase access to this region.

The EC can also access any other flash region as permitted by the Flash Descriptor settings. As such, the EC uses linear addresses, valid up to the maximum supported flash size, to access the flash.

The MAFCC supports flash read, write, and erase operations only.

16.6.9.4.2 Slave Attached Flash Channel Controller (SAFCC) Flash Operation and Addressing

The PCH is allocated dedicated regions (for each of the supported masters) within the eSPI slave-attached flash devices. The PCH has read, write, and erase access to these regions, as well as any other regions that maybe permitted by the region protections set in the Flash Descriptor.

The Slave will optionally performs additional checking on the PCH provided address. In case of an error due to incorrect address or any other issues it will synthesize an unsuccessful completion back to the eSPI Master.

The SAFCC supports Flash Read, Write and Erase operations. It also supports Read SFDP and Read JEDEC ID commands as specified in the eSPI Specification for Server platforms.

§ §

17 General Purpose Input and Output (GPIO)

17.1 Acronyms

Acronyms	Description
GPI	General Purpose Input
GPO	General Purpose Output
GPP	General Purpose I/O in Primary Well
GPD	General Purpose I/O in Deep Sleep Well

17.2 Overview

The PCH General Purpose Input/Output (GPIO) signals are grouped into multiple groups (such as GPP_A, GPP_B, and so on) and are powered by either the PCH Primary well or Deep Sleep well.

The high level features of GPIO:

- Configurable 3.3V or 1.8V voltage (except for GPP F and GPD groups)
- Configurable as an input or output signal.
- Configurable GPIO pad ownership by host, ME, or ISH.
- SCI (GPE) and interrupt capable on all GPIOs
- NMI and SMI capability capable (on selected GPIOs).
- PWM, Serial Blink capable (on selected GPIOs).
- Programmable hardware debouncer (on GPD3/PWRBTN# pin)

17.3 Signal Description

Table 17-1 summarizes the GPIO implementation in the PCH.

Table 17-1. General Purpose I/O Signals (Sheet 1 of 9)

GPIO	Native Function 1	Native Dir 1	Native Function 2	Native Dir 2	Native Function 3	Native Dir 3	Native Function 4	Native Dir	Native Function 5	Native Dir	Default	Strap?	Input Deglitch	Output Power Sequence Deglitch	Pin State During Reset	Pin State Immediately After Reset	Notes
Primary Well Group A (Per-Pad 1.8 V or 3.3 V)																	
GPP_A0	ESPI_IO0	inout									Native F1/GP-In		no	yes(1)	Native F1: PU / GPIO: Z	Native F1: PU/ GPIO: Z	Native F1 if eSPI Disable pin strap = 0, else GP-In
GPP_A1	ESPI_IO1	inout									Native F1/GP-In		no	yes(1)	Native F1: PU / GPIO: Z	Native F1: PU / GPIO: Z	Native F1 if eSPI Disable pin strap = 0, else GP-In
GPP_A2	ESPI_IO2	inout									Native F1/GP-In		no	yes(1)	Native F1: PU / GPIO: Z	Native F1: PU / GPIO: Z	Native F1 if eSPI Disable pin strap = 0, else GP-In

Table 17-1. General Purpose I/O Signals (Sheet 2 of 9)

GPIO	Native Function 1	Native Dir 1	Native Function 2	Native Dir 2	Native Function 3	Native Dir 3	Native Function 4	Native Dir	Native Function 5	Native Dir	Default	Strap?	Input Deglitch	Output Power Sequence Deglitch	Pin State During Reset	Pin State Immediately After Reset	Notes
GPP_A3	ESPI_IO3	inout									Native F1/GP-In		no	yes(1)	Native F1: PU / GPIO: Z	Native F1: PU / GPIO: Z	Native F1 if eSPI Disable pin strap = 0, else GP-In
GPP_A4	ESPI_CS#	out									Native F1/GP-In		no	yes(2)	Native F1: PU / GPIO: Z	Native F1: H / GPIO: Z	Native F1 if eSPI Disable pin strap = 0, else GP-In
GPP_A5	ESPI_CLK	out									Native F1/GP-In		no	yes(1)	Native F1: PD / GPIO: Z	Native F1: L / GPIO: Z	Native F1 if eSPI Disable pin strap = 0, else GP-In
GPP_A6	ESPI_RESET#	out									Native F1/GP-In		no	yes(1)	Native F1: L / GPIO: Z	Native F1: H / GPIO: Z	Native F1 if eSPI Disable pin strap = 0, else GP-In
GPP_A7	I2S2_SCLK	inout									GP-In		no	yes(1)	Z	Z	
GPP_A8	I2S2_SFRM	inout	CNV_RF_RESET#	out							Native F2		no	yes(2)	L	L	
GPP_A9	I2S2_TXD	out	MODEM_CLKREQ	out							Native F2		no	yes(1)	Z	Z	
GPP_A10	I2S2_RXD	in									GP-In		no	yes(1)	Z	Z	
GPP_A11	SATA_DEVSLP2	od									GP-In		no	yes(1)	Z	Z	
GPP_A12	SATAXPCIE1	in	SATAGP1	in							Native F1/GP-In		no	yes(1)	Native F1: PU / GPIO: Z	Native F1: PU / GPIO: Z	Native F1 if SATA / PCIe GP Select for Port 1 soft strap = 00b or 01b; if soft strap = 11b GP-In. **Note**: Different Connectors have different polarity for the SATAxPCIe select. **Note**: Default state is GP-In before soft straps are loaded or in the case the soft straps are invalid
GPP_A13	SATAXPCIE2	in	SATAGP2	in							Native F1/GP-In		no	yes(1)	Native F1: PU / GPIO: Z	Native F1: PU / GPIO: Z	Native F1 if SATA / PCIe GP Select for Port 2 soft strap = 00b or 01b; if soft strap = 11b GP-In. **Note**: Different Connectors have different polarity for the SATAxPCIe select. **Note**: Default state is GP-In before soft straps are loaded or in the case the soft straps are invalid
GPP_A14	USB_OC1#	in	DDSP_HPD3	in			DISP_MISC3	out			GP-In		no	yes(1)	Z	Z	
GPP_A15	USB_OC2#	in	DDSP_HPD4	in			DISP_MISC4	out			GP-In		no	yes(1)	Z	Z	
GPP_A16	USB_OC3#	in									GP-In		no	yes(1)	Z	Z	
GPP_A17			DISP_MISCC	out							GP-In		no	yes(1)	Z	Z	
GPP_A18	DDSP_HPDB	in	DISP_MISCB	out							GP-In		no	yes(1)	Z	Z	
GPP_A19	DDSP_HPD1	in	DISP_MISC1	out							GP-In		no	yes(1)	Z	Z	
GPP_A20	DDSP_HPD2	in	DISP_MISC2	out							GP-In		no	yes(1)	Z	Z	
GPP_A21											GP-In		no	yes(1)	Z	Z	
GPP_A22											GP-In		no	yes(1)	Z	Z	
GPP_A23	I2S1_SCLK	inout									GP-In		no	yes(1)	Z	Z	
Primary Well Group B (Per-Pad 1.8 V or 3.3 V)																	
GPP_B0	CORE_VID0	out									Native F1		no	yes(1)	H or L	H or L	
GPP_B1	CORE_VID1	out									Native F1		no	yes(1)	H or L	H or L	
GPP_B2	VRALERT#	in									GP-In		no	yes(1)	Z	Z	
GPP_B3	CPU_GP2	in									GP-In		no	yes(1)	Z	Z	
GPP_B4	CPU_GP3	in									GP-In		no	yes(1)	Z	Z	

Table 17-1. General Purpose I/O Signals (Sheet 3 of 9)

GPIO	Native Function 1	Native Dir 1	Native Function 2	Native Dir 2	Native Function 3	Native Dir 3	Native Function 4	Native Dir	Native Function 5	Native Dir	Default	Strap?	Input Deglitch	Output Power Sequence Deglitch	Pin State During Reset	Pin State Immediately After Reset	Notes
GPP_B5	ISH_I2C0_SDA	iod									GP-In		no	yes(1)	Z	Z	
GPP_B6	ISH_I2C0_SCL	iod									GP-In		no	yes(1)	Z	Z	
GPP_B7	ISH_I2C1_SDA	iod									GP-In		no	yes(1)	Z	Z	
GPP_B8	ISH_I2C1_SCL	iod									GP-In		no	yes(1)	Z	Z	
GPP_B9	I2C5_SDA	iod	ISH_I2C2_SDA	iod							GP-In		no	yes(1)	Z	Z	
GPP_B10	I2C5_SCL	iod	ISH_I2C2_SCL	iod							GP-In		no	yes(1)	Z	Z	
GPP_B11	PMCALERT#	iod									GP-In		no	yes(1)	Z	Z	
GPP_B12	SLP_S0#	out									Native F1		no	yes(1)	H	H	
GPP_B13	PLTRST#	out									Native F1		no	yes(2)	L	H	
GPP_B14	SPKR	out	TIME_SYNC1	inout	GSPI0_CS1#	out					GP-Out	Top Swap Override	no	no	PD	L	Strap read at rising edge of PCH_PWROK. The internal 20 kohm ± 30% Pull-down is disabled after PCH_PWROK is high.
GPP_B15	GSPI0_CS0#	out									GP-In		no	yes(1)	Z	Z	
GPP_B16	GSPI0_CLK	out									GP-In		no	yes(1)	Z	Z	
GPP_B17	GSPI0_MISO	in									GP-In		no	yes(1)	Z	Z	
GPP_B18	GSPI0_MOSI	out									GP-Out	No Reboot	no	no	PD	Z	Strap read at rising edge of PCH_PWROK. The internal 20 kohm ± 30% Pull-down is disabled after PCH_PWROK is high.
GPP_B19	GSPI1_CS0#	out									GP-In		no	yes(1)	Z	Z	
GPP_B20	GSPI1_CLK	out		out							GP-In		no	yes(1)	Z	Z	
GPP_B21	GSPI1_MISO	in		in							GP-In		no	yes(1)	Z	Z	
GPP_B22	GSPI1_MOSI	out									GP-In		no	no	Z	Z	
GPP_B23	SML1ALERT#	iod	PCHHOT#	od	GSPI1_CS1#	out					GP-Out	CPUNSSC Clock Frequency	yes	no	PD	Z	Strap read at rising edge of RSMRST#. The internal 20 kohm ± 30% Pull-down is disabled after RSMRST# de-asserts. Note: When used as PCHHOT# and strap low, a 150k weak board Pull-up is recommended to ensure it does not override the internal Pull-down strap sampling.
Primary Well Group C (Per-Pad 1.8 V or 3.3 V)																	
GPP_C0	SMBCLK	iod									Native F1		yes	yes(1)	Z	Z	
GPP_C1	SMBDATA	iod									Native F1		yes	yes(1)	Z	Z	
GPP_C2	SMBALERT#	iod									Native F1/GP-Out	TLS Confidentiality	yes	no	Native F1: PD / GPIO: PD	Native F1: L / GPIO: Z	Native F1 if Intel® SMBus ASD Mode Configuration soft strap = 1, else GP-Out; Strap read at rising edge of RSMRST#. The internal 20 kohm ± 30% Pull-down is disabled after RSMRST# de-asserts.
GPP_C3	SML0CLK	iod									Native F1		yes	yes(1)	Z	Z	
GPP_C4	SML0DATA	iod									Native F1		yes	yes(1)	Z	Z	
GPP_C5	SML0ALERT#	iod									GP-Out	eSPI Disable	yes	no	PD	Z	Strap read at rising edge of RSMRST#. The internal 20 kohm ± 30% Pull-down is disabled after RSMRST# de-asserts.

Table 17-1. General Purpose I/O Signals (Sheet 4 of 9)

GPIO	Native Function 1	Native Dir 1	Native Function 2	Native Dir 2	Native Function 3	Native Dir 3	Native Function 4	Native Dir	Native Function 5	Native Dir	Default	Strap?	Input Deglitch	Output Power Sequence Deglitch	Pin State During Reset	Pin State Immediately After Reset	Notes
GPP_C6	SML1CLK	iod	SUSWARN# / SUSPWRDNACK	out							Native F2/GP-In		yes	yes(1)	Native F2: L / GPIO: Z	Native F2: L / GPIO: Z	Native F2 if eSPI Disable pin strap = 0, else GP-In
GPP_C7	SML1DATA	iod	SUSACK#	in							Native F2/GP-In		yes	yes(1)	Native F2: H / GPIO: Z	Native F2: H / GPIO: Z	Native F2 if eSPI Disable pin strap = 0, else GP-In
GPP_C8	UART0_RXD	in		in out							GP-In		no	yes(1)	Z	Z	
GPP_C9	UART0_TXD	out		out							GP-In		no	yes(1)	Z	Z	
GPP_C10	UART0_RTS#	out		out							GP-In		no	yes(1)	Z	Z	
GPP_C11	UART0_CTS#	in	CNV_BT_UART_CTS#	in							GP-In		no	yes(1)	Z	Z	
GPP_C12	UART1_RXD	in	ISH_UART1_RXD	in							GP-In		no	yes(1)	Z	Z	
GPP_C13	UART1_TXD	out	ISH_UART1_TXD	out							GP-In		no	yes(1)	Z	Z	
GPP_C14	UART1_RTS#	out	ISH_UART1_RTS#	out							GP-In		no	yes(1)	Z	Z	
GPP_C15	UART1_CTS#	in	ISH_UART1_CTS#	in							GP-In		no	yes(1)	Z	Z	
GPP_C16	I2C0_SDA	iod									GP-In		yes	yes(1)	Z	Z	
GPP_C17	I2C0_SCL	iod									GP-In		yes	yes(1)	Z	Z	
GPP_C18	I2C1_SDA	iod									GP-In		yes	yes(1)	Z	Z	
GPP_C19	I2C1_SCL	iod									GP-In		yes	yes(1)	Z	Z	
GPP_C20	UART2_RXD	in									GP-In		no	yes(1)	Z	Z	
GPP_C21	UART2_TXD	out									GP-In		no	yes(1)	Z	Z	
GPP_C22	UART2_RTS#	out									GP-In		no	yes(1)	Z	Z	
GPP_C23	UART2_CTS#	in									GP-In		no	yes(1)	Z	Z	
Primary Well Group D (Per-Pad 1.8 V or 3.3 V)																	
GPP_D0	ISH_GP0	inout									GP-In		no	yes(1)	Z	Z	
GPP_D1	ISH_GP1	inout									GP-In		no	yes(1)	Z	Z	
GPP_D2	ISH_GP2	inout									GP-In		no	yes(1)	Z	Z	
GPP_D3	ISH_GP3	inout									GP-In		no	yes(1)	Z	Z	
GPP_D4	IMGCLKOUT0	out									GP-In		no	yes(1)	Z	Z	
GPP_D5	SRCCLKREQ0#	iod									GP-In		yes	yes(1)	Z	Z	
GPP_D6	SRCCLKREQ1#	iod									GP-In		yes	yes(1)	Z	Z	
GPP_D7	SRCCLKREQ2#	iod									GP-In		yes	yes(1)	Z	Z	
GPP_D8	SRCCLKREQ3#	iod									GP-In		yes	yes(1)	Z	Z	
GPP_D9	ISH_SPI_CS#	out	DDP3_CTRLCLK	inout	GSPI2_CS0#	out	TBT_LSX2_TXD	inout	Reserved	inout	Native F5		no	yes(1)	Z	Z	
GPP_D10	ISH_SPI_CLK	out	DDP3_CTRLDATA	inout	GSPI2_CLK	out	TBT_LSX2_RXD	inout	Reserved	inout	Native F5	DDP3 I2C / TBT_LSX2 pins VCC configuration	no	yes(1)	Z	Z	Strap read at rising edge of RSMRST#. External pull-down is required if signal used as HDMI Display I2C.
GPP_D11	ISH_SPI_MISO	in	DDP4_CTRLCLK	inout	GSPI2_MISO	in	TBT_LSX3_TXD	inout	Reserved	inout	Native F5		no	no	Z	Z	
GPP_D12	ISH_SPI_MOSI	out	DDP4_CTRLDATA	inout	GSPI2_MOSI	out	TBT_LSX3_RXD	inout	Reserved	inout	Native F5	DDP4 USB-C*I2C/ TBT_LSX3 pins VCC configuration	no	no	Z	Z	Strap read at rising edge of RSMRST#. External pull-down is required if signal used as HDMI Display I2C.

Table 17-1. General Purpose I/O Signals (Sheet 5 of 9)

GPIO	Native Function 1	Native Dir 1	Native Function 2	Native Dir 2	Native Function 3	Native Dir 3	Native Function 4	Native Dir	Native Function 5	Native Dir	Default	Strap?	Input Deglitch	Output Power Sequence Deglitch	Pin State During Reset	Pin State Immediately After Reset	Notes
GPP_D13	ISH_UART 0_RXD	in						in			GP-In		no	yes(1)	Z	Z	
GPP_D14	ISH_UART 0_TXD	out									GP-In		no	yes(1)	Z	Z	
GPP_D15	ISH_UART 0_RTS#	out	GSPI2_CS1 #	out	IMGCLKOU T5	out					GP-In		no	yes(1)	Z	Z	
GPP_D16	ISH_UART 0_CTS#	in						in			GP-In		no	yes(1)	Z	Z	
GPP_D17	ISH_GP4	inout									GP-In		no	yes(1)	Z	Z	
GPP_D18	ISH_GP5	inout									GP-In		no	yes(1)	Z	Z	
GPP_D19	I2S_MCLK	out									GP-In		no	yes(1)	Z	Z	
Primary Well Group E (Per-Pad 3.3 V or 1.8 V)																	
GPP_E0	SATAXPCI E0	in	SATAGP0	in							Native F1/GP-In		no	yes(1)	Native F1: PU / GPIO: Z	Native F1: PU / GPIO: Z	Native F1 if SATA / PCIe GP Select for Port 0 soft strap = 00b or 01b; if soft strap = 11b GP-In. **Note**: Different Connectors have different polarity for the SATAxPCIe select. **Note**: Default state is GP-In before soft straps are loaded or in the case the soft straps are invalid
GPP_E1	SPI1_IO2	inout									GP-In		no	yes(1)	Z	Z	
GPP_E2	SPI1_IO3	inout									GP-In		no	yes(1)	Z	Z	
GPP_E3	CPU_GP0	in									GP-In		no	yes(1)	Z	Z	
GPP_E4	SATA_DEV SLP0	od									GP-In		no	yes(1)	Z	Z	
GPP_E5	SATA_DEV SLP1	od									GP-In		no	yes(1)	Z	Z	
GPP_E6											GP-Out	Reserved	no	yes(1)	Z	Z	Strap read at rising edge of RSMRST#. External pull-up is required. Recommend 100K. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.
GPP_E7	CPU_GP1	in									GP-In		no	yes(1)	Z	Z	
GPP_E8	SATA_LED #	od	SPI1_CS1 #	out							GP-In		no	yes(1)	Z	Z	
GPP_E9	USB_OC0 #	in									GP-In		no	yes(1)	Z	Z	
GPP_E10	SPI1_CS#	out	BK0						SBK0		GP-In		no	yes(1)	Z	Z	
GPP_E11	SPI1_CLK	out	BK1						SBK1		GP-In		no	yes(1)	Z	Z	
GPP_E12	SPI1_MIS O	inout	BK2						SBK2		GP-In		no	yes(1)	Z	Z	
GPP_E13	SPI1_MOS I	inout	BK3						SBK3		GP-In		no	yes(1)	Z	Z	
GPP_E14	DDSP_HP DA	in	DISP_MISC A	out							GP-In		no	yes(1)	Z	Z	
GPP_E15	ISH_GP6	inout	Reserved								Native F2		no	yes(1)	Z	Z	
GPP_E16	ISH_GP7	inout	Reserved								Native F2		no	yes(1)	Z	Z	
GPP_E17											GP-In		no	yes(1)	Z	Z	
GPP_E18	DDP1_CTR LCLK	inout					TBT_LSX0 _TXD	inout	Reserved	inout	Native F5		yes	yes(1)	Z	Z	
GPP_E19	DDP1_CTR LDATA	inout					TBT_LSX0 _RXD	inout	Reserved	inout	Native F5	DDP1 I2C / TBT_LSX0 pins VCC configuration	yes	no	Z	Z	Strap read at rising edge of RSMRST#. External pull-up is required if signal used as HDMI Display I2C.
GPP_E20	DDP2_CTR LCLK	inout					TBT_LSX1 _TXD	inout	Reserved	inout	Native F5		yes	yes(1)	Z	Z	

Table 17-1. General Purpose I/O Signals (Sheet 6 of 9)

GPIO	Native Function 1	Native Dir 1	Native Function 2	Native Dir 2	Native Function 3	Native Dir 3	Native Function 4	Native Dir	Native Function 5	Native Dir	Default	Strap?	Input Deglitch	Output Power Sequence Deglitch	Pin State During Reset	Pin State Immediately After Reset	Notes
GPP_E21	DDP2_CTRLDATA	inout					TBT_LSX1_RXD	inout	Reserved	inout	Native F5	DDP2 I2C/ TBT_LSX1 pins VCC configuration	yes	no	PD	Z	Strap read at rising edge of RSMRST#. The internal 20 kohm ± 30% Pull-down is disabled after RSMRST# de-asserts. External pull-up is required if signal used as HDMI Display I2C.
GPP_E22	DDPA_CTRLCLK	inout			PCIE_LNK_DOWN	out					GP-In		yes	yes(1)	Z	Z	
GPP_E23	DDPA_CTRLDATA	inout	BK4						SBK4		GP-Out		yes	no	PD	Z	
Primary Well Group F (1.8 V Only)																	
GPP_F0	CNV_BRI_DT	out	UART0_RTS#	out							Native F1	XTAL Frequency Selection	no	yes(1)	PD	H	Strap read at rising edge of RSMRST#. The internal 20 kohm ± 30% pull-down is disabled after RSMRST# de-asserts. This strap should not be pulled high since 24 MHz crystal is not supported.
GPP_F1	CNV_BRI_RSP	in	UART0_RXD	in							Native F1		no	yes(1)	Z	H	
GPP_F2	CNV_RGI_DT	out	UART0_TXD	out							Native F1	M.2 CNVi Mode Select	no	yes(1)	Z	H	A weak external PU is requires to disable CNVi by default. When a companion chip is connected, the required PD on the device will pull the strap low to enable CNVi.
GPP_F3	CNV_RGI_RSP	in	UART0_CTS#	in							Native F1		no	yes(1)	Z	H	
GPP_F4	CNV_RF_RESET#	out									Native F1		no	yes(2)	PD	H	
GPP_F5			MODEM_CLKREQ	out							Native F2		no	yes(1)	Z	H	
GPP_F6	CNV_PA_BLANKING	in									GP-In		no	yes(1)	Z	Z	
GPP_F7	EMMC_CMD	inout									GP-In		no	yes(1)	Z	Z	
GPP_F8	EMMC_DATA0	inout									GP-In		no	yes(1)	Z	Z	
GPP_F9	EMMC_DATA1	inout									GP-In		no	yes(1)	Z	Z	
GPP_F10	EMMC_DATA2										GP-In		no	yes(1)	Z	Z	
GPP_F11	EMMC_DATA3	inout									GP-In		no	yes(1)	Z	Z	
GPP_F12	EMMC_DATA4	inout									GP-In		no	yes(1)	Z	Z	
GPP_F13	EMMC_DATA5	inout									GP-In		no	yes(1)	Z	Z	
GPP_F14	EMMC_DATA6	inout									GP-In		no	yes(1)	Z	Z	
GPP_F15	EMMC_DATA7	inout									GP-In		no	yes(1)	Z	Z	
GPP_F16	EMMC_RCLK	in									GP-In		no	yes(1)	Z	Z	
GPP_F17	EMMC_CLK	out									GP-In		no	yes(1)	Z	Z	
GPP_F18	EMMC_RESET#	out									GP-In		no	yes(1)	Z	Z	
GPP_F19	A4WP_PRESENT	in									Native F1/GP-in		no	yes(1)	Z	L	
Primary Well Group G (Per-pad 1.8 V or 3.3 V)																	
GPP_G0	SD_CMD	inout									GP-In		no	yes(1)	Z	Z	
GPP_G1	SD_DATA0	inout									GP-In		no	yes(1)	Z	Z	
GPP_G2	SD_DATA1	inout									GP-In		no	yes(1)	Z	Z	
GPP_G3	SD_DATA2	inout									GP-In		no	yes(1)	Z	Z	

Table 17-1. General Purpose I/O Signals (Sheet 7 of 9)

GPIO	Native Function 1	Native Dir 1	Native Function 2	Native Dir 2	Native Function 3	Native Dir 3	Native Function 4	Native Dir	Native Function 5	Native Dir	Default	Strap?	Input Deglitch	Output Power Sequence Deglitch	Pin State During Reset	Pin State Immediately After Reset	Notes
GPP_G4	SD_DATA3	inout									GP-In		no	yes(1)	Z	Z	
GPP_G5	SD_CD#	in									GP-In		no	yes(1)	Z	Z	
GPP_G6	SD_CLK	out									GP-In		no	yes(1)	Z	Z	
GPP_G7	SD_WP	in									GP-In		no	yes(1)	Z	Z	
Primary Well Group H (Per-pad 1.8 V or 3.3 V)																	
GPP_H0			CNV_BT_I2S_SCLK	inout							GP-In		no	yes(1)	Z	Z	
GPP_H1	SD_PWR_EN#	out	CNV_BT_I2S_BCLK	inout							GP-In		no	yes(1)	Z	Z	
GPP_H2			CNV_BT_I2S_SDI	in							GP-Out	eSPI Flash Sharing Mode	no	yes(1)	PD	Z	Strap read at rising edge of RSMRST#. The internal 20 kohm ± 30% pull-down is disabled after RSMRST# de-asserts.
GPP_H3	SX_EXIT_HOLDOFF#	in	CNV_BT_I2S_SDO	out							GP-In		no	yes(1)	Z	Z	
GPP_H4	I2C2_SDA	iod									GP-In		yes	yes(1)	Z	Z	
GPP_H5	I2C2_SCL	iod									GP-In		yes	yes(1)	Z	Z	
GPP_H6	I2C3_SDA	iod									GP-In		yes	yes(1)	Z	Z	
GPP_H7	I2C3_SCL	iod									GP-In		yes	yes(1)	Z	Z	
GPP_H8	I2C4_SDA	iod	CNV_MFUART2_RXD	in							GP-In		yes	yes(1)	Z	Z	
GPP_H9	I2C4_SCL	iod	CNV_MFUART2_TXD	out							GP-In		yes	yes(1)	Z	Z	
GPP_H10	SRCCLKREQ4#	iod									GP-In		yes	yes(1)	Z	Z	
GPP_H11	SRCCLKREQ5#	iod									GP-In		yes	yes(1)	Z	Z	
GPP_H12	M2_SKT2_CFG0	in									GP-In		no	yes(1)	Z	Z	
GPP_H13	M2_SKT2_CFG1	in									GP-In		no	yes(1)	Z	Z	
GPP_H14	M2_SKT2_CFG2	in									GP-In		no	yes(1)	Z	Z	
GPP_H15	M2_SKT2_CFG3	in									GP-In		no	yes(1)	Z	Z	
GPP_H16	DDPB_CTRLCLK	inout									GP-In		no	yes(1)	Z	Z	
GPP_H17	DDPB_CTRLDATA	inout									GP-Out		no	no	PD	Z	The internal 20 kohm ± 30% pull-down is disabled after PCH_PWROK is high.
GPP_H18	CPU_C10_GATE#	out									Native F1		no	yes(1)	H (Refer note)	H	The During Reset Pin State of this GPIO is H via a ~20K pull-up to 3.3 V independent of soft strap assigned pad voltage. A 1.8 V device connected to this GPIO must be capable of taking ~20K pull-up to 3.3 V.
GPP_H19	TIME_SYNC0	inout									GP-In		no	yes(1)	Z	Z	
GPP_H20	IMGCLKOUT1	out									GP-In		no	yes(1)	Z	Z	
GPP_H21	IMGCLKOUT2	out									GP-In		no	no	PD	Z	
GPP_H22	IMGCLKOUT3	out									GP-In		no	no	Z	Z	
GPP_H23	IMGCLKOUT4	out									GP-In		no	no	PD	L	
Primary Well Group R (Per-family 1.8 V or 3.3 V)																	
GPP_R0	HDA_BCLK	inout	I2S0_SCLK	inout							Native F1		no	yes(1)	L	L	
GPP_R1	HDA_SYNC	inout	I2S0_SFRM	inout							Native F1		no	yes(1)	PD	L	

Table 17-1. General Purpose I/O Signals (Sheet 8 of 9)

GPIO	Native Function 1	Native Dir 1	Native Function 2	Native Dir 2	Native Function 3	Native Dir 3	Native Function 4	Native Dir	Native Function 5	Native Dir	Default	Strap?	Input Deglitch	Output Power Sequence Deglitch	Pin State During Reset	Pin State Immediately After Reset	Notes
GPP_R2	HDA_SDO	inout	I2S0_TXD	out							Native F1	Flash Descriptor Security Override	no	yes(1)	PD	Z	Strap read at rising edge of PCH_PWROK. The internal 20 kohm ± 30% Pull-down is disabled after PCH_PWROK is high.
GPP_R3	HDA_SDI0	inout	I2S0_RXD	in							Native F1		no	yes(1)	PD	PD	
GPP_R4	HDA_RST#	inout									Native F1		no	yes(1)	L	L	
GPP_R5	HDA_SDI1	inout	I2S1_SFRM	inout							GP-In		no	yes(1)	Z	Z	
GPP_R6			I2S1_TXD	out							GP-In		no	yes(1)	Z	Z	
GPP_R7			I2S1_RXD	in							GP-In		no	yes(1)	Z	Z	
Primary Well Group S (1.8 V Only)																	
GPP_S0	SNDW1_CLK	inout									GP-In		no	yes(1)	Z	Z	
GPP_S1	SNDW1_DATA	inout									GP-In		no	yes(1)	Z	Z	
GPP_S2	SNDW2_CLK	inout									GP-In		no	yes(1)	Z	Z	
GPP_S3	SNDW2_DATA	inout									GP-In		no	yes(1)	Z	Z	
GPP_S4	SNDW3_CLK	inout	DMIC_CLK1	out							GP-In		no	yes(1)	Z	Z	
GPP_S5	SNDW3_DATA	inout	DMIC_DATA1	in							GP-In		no	yes(1)	Z	Z	
GPP_S6	SNDW4_CLK	inout	DMIC_CLK0	out							GP-In		no	yes(1)	Z	Z	
GPP_S7	SNDW4_DATA	inout	DMIC_DATA0	in							GP-In		no	yes(1)	Z	Z	
Deep Sleep Well Group (3.3 V Only)																	
GPD0	BATLOW#	in									Native F1		yes	yes(1)	Z	Z	
GPD1	ACPRESENT	in									Native F1		yes	yes(1)	Refer note	Z	Pin state during reset: In Deep Sx enabled configurations, pin is PD when PD is enabled (Refer ACPRES_PD_DSX_DIS bit); else Z
GPD2	LAN_WAKE#	in									Native F1		yes	yes(1)	Z	Z	
GPD3	PWRBTN#	in									Native F1		yes	yes(1)	PU	PU	
GPD4	SLP_S3#	out									Native F1/GP-Out		no	yes(2)	Native F1: L / GPIO: L	Native F1: H / GPIO: L	Native if SLP_S3# / GPD4 Signal Configuration soft strap = 0, else GP-out
GPD5	SLP_S4#	out									Native F1/GP-Out		no	yes(2)	Native F1: L / GPIO: L	Native F1: H / GPIO: L	Native if SLP_S4# / GPD5 Signal Configuration soft strap = 0, else GP-out
GPD6	SLP_A#	out									Native F1/GP-Out		no	yes(2)	Native F1: L / GPIO: L	Native F1: H / GPIO: L	Native if SLP_A# / GPD6 Signal Configuration soft strap = 0, else GP-out
GPD7											GP-Out	Reserved	no	no	PD	L	Strap read at rising edge of DSW_PWROK. The internal pull-down 20 kohm ± 30% is disabled after DSW_PWROK is high . This strap should sample LOW. There should NOT be any on-board device driving it to opposite direction during strap sampling.
GPD8	SUSCLK	out									Native F1		no	yes(2)	L	T	
GPD9	SLP_WLAN#	out									Native F1/GP-Out		no	yes(2)	Native F1: L / GPIO: L	Native F1: L / GPIO: L	Native if SLP_WLAN# / GPD9 Signal Configuration soft strap = 0, else GP-out
GPD10	SLP_S5#	out									Native F1/GP-Out		no	yes(2)	Native F1: L / GPIO: L	Native F1: H / GPIO: L	Native if SLP_S5# / GDP10 Signal Configuration soft strap = 0, else GP-out
GPD11	LANPHYPC	out									Native F1/GP-Out		no	yes(2)	Native F1: L / GPIO: L	Native F1: L / GPIO: Z	Native if LAN PHY Power Control GPD11 Signal Configuration soft strap = 0, else GP-out

Table 17-1. General Purpose I/O Signals (Sheet 9 of 9)

GPIO	Native Function 1	Native Dir 1	Native Function 2	Native Dir 2	Native Function 3	Native Dir 3	Native Function 4	Native Dir	Native Function 5	Native Dir	Default	Strap?	Input Deglitch	Output Power Sequence Deglitch	Pin State During Reset	Pin State Immediately After Reset	Notes
Deglitch Legend yes(1) - o/p Hi-Z, no internal weak pull during respective pin power sequencing yes(2) - o/p Hi-Z, with integrated 20 kohm ± 30% pull-down during respective pin power sequencing yes(3) - o/p Hi-Z, with integrated 20 kohm ± 30% pull-up during respective pin power sequence													**Pin State Legend** H- Driven High L - Driven Low Z - Hi-Z PU - Integrated 20 kohm ± 30% pull-up pulls pin high PD - Integrated 20 kohm ± 30% pull-down pulls pin low T - Toggling				

17.4 Functional Description

17.4.1 Configurable GPIO Voltage

Except for all pads in GPIO F, GPIO S, and GPD groups, all other GPIO pads support per-pad configurable voltage, which allows control selection of 1.8V or 3.3V for each pad. The configuration is done via soft straps.

Before soft straps are loaded, the default voltage of each pin depends on its default as input or output.

- Input: 1.8V level with 3.3V tolerant.
- Output: the pin drives 3.3V via a ~20K pull-up. With this, any 1.8V device must be capable of taking 20K pull-up to 3.3V.

Warning: GPIO pad voltage configuration must be set correctly depending on device connected to it; otherwise, damage to the PCH or the device may occur.

Note: GPIO F and S groups support 1.8V only.

Note: GPD group supports 3.3V only.

17.4.2 GPIO Buffer Impedance Compensation via SD3_RCOMP

All GPIO buffers require impedance compensation for 1.8V and 3.3V operation. The impedance compensation is done via the SD3_RCOMP signal. Therefore, SD3_RCOMP signal must have a precision pull down resistor of 200 Ohm (1%) to GND (regardless of SDXC being used or not). Without proper impedance compensation, the GPIO buffers, including the muxed native functions, may not operate as expected.

17.4.3 Programmable Hardware Debouncer

Hardware debounce capability is supported on GPD3/PWRBTN# pad. The capability can be used to filter signal from switches and buttons if needed.

The period can be programmed from 8 to 32768 times of the RTC clock by programming the Pad Configuration DW2 register. At 32 kHz RTC clock, the debounce period is 244us to 1s.

17.4.4 Integrated Pull-ups and Pull-downs

All GPIOs have programmable internal pull-up/pull-down resistors which are off by default. The internal pull-up/pull-down for each GPIO can be enabled by BIOS programming the corresponding PAD_CFG_DW0 register. Refer to Volume 2 (Register Info) for more details.

17.4.5 SCI / SMI# and NMI

SCI capability is available on all GPIOs, while SMI and NMI capability is available on only select GPIOs.

Below are the PCH GPIOs that can be routed to generate SMI# or NMI:

- GPP_B14, GPP_B20, GPP_B23
- GPP_C[23:22]
- GPP_D[4:0]
- GPP_E[8:0], GPP_E[16:13]

17.4.6 Timed GPIO (TIME_SYNC)

The PCH supports 2 Timed GPIOs as native function (TIME_SYNC) that is muxed on GPIO pins. The intent usage of the Timed GPIO function is for time synchronization purpose.

Timed GPIO can be an input or an output.

- As an input, a GPIO input event triggers the HW to capture the PCH Always Running Timer (ART) time in the Time Capture register. The GPIO input event must be asserted for at least 2 crystal oscillator clocks period in order for the event to be recognized.
- As an output, a match between the ART time and the software programmed time value triggers the HW to generate a GPIO output event and capture the ART time in the Time Capture register. If periodic mode is enabled, HW generates the periodic GPIO events based on the programmed interval. The GPIO output event is asserted by HW for at least 2 crystal oscillator clocks period.

Timed GPIO supports event counter. When Timed GPIO is configured as input, event counter increments by 1 for every input event triggered. When Timed GPIO is configured as output, event counter increments by 1 for every output event generated. The event counter provides the correlation to associate the Timed GPIO event (the nth event) with the captured ART time. The event counter value is captured when a read to the Time Capture Value register occurs.

Note: When Timed GPIO is enabled, the crystal oscillator will not be shut down as crystal clock is needed for the Timed GPIO operation. As a result, SLP_S0# will not be asserted. This has implication to platform power (such as IDLE or S0ix power). Software should only enable Timed GPIO when needed and disable it when Timed GPIO functionality is not required.

17.4.7 GPIO Blink (BK) and Serial Blink (SBK)

Certain GPIOs are capable of supporting blink and serial blink, indicated as BK and SBK respectively in the GPIO Signals table above. The BK and SBK are implemented as native functions muxed on the selected GPIOs. To enable BK or SBK on a GPIO having the capability, BIOS needs to select the assigned native function for BK or SBK on the GPIO.

17.4.8 Interrupt / IRQ via GPIO Requirement

A GPIO, as an input, can be used to generate an interrupt/IRQ to the PCH. In this case, it is required that the pulse width on the GPIO must be at least 4 us for the PCH to recognize the interrupt.

17.4.9 GPIO Ownership

Any PCH GPIO can be owned either by the host or the Intel® CSME. The designer can select GPIOs that are required by a ME feature using the ME FIT tool (available with Intel ME FW releases). When selected and controlled by the ME, those GPIOs cannot be used by the host anymore.

17.4.10 Native Function and TERM Bit Setting

Certain native function signals that are muxed onto GPIO pins support dynamic termination override, which allows the native controller to dynamically control the integrated pull-up / pull-down resistors on the signals. For those native function signals, when used, software must program the TERM bit field in the corresponding GPIO's Pad Configuration DW1 to 1111b. Refer to Volume 2 for information on the PAD configuration DW1 registers and the TERM bit field.

The table below shows the native function signals that support dynamic termination override.

Table 17-2. Native Function Signals Supporting Dynamic Termination Override

Native Function	Signal with Dynamic Termination Override
Intel® HD Audio	HDA_SDI[0:1] HDA_SDO HDA_SYNC I2S[2:0]_SCLK I2S[2:0]_SFRM I2S[2:0]_RXD DMIC_DATA[1:0] SNDW_DATA[4:1]
SPI1	SPI1_MOSI SPI1_MISO SPI1_IO[3:2]
SDXC (SD Card)	SD_CMD SD_DATA[3:0] SD_CLK

17.4.11 Virtual GPIO (vGPIO)

vGPIO is a special type of GPIO implemented in the PCH for a specific functionality. vGPIO is not a physical GPIO; the signal is not balled out on the package. Programming the vGPIO is similar to programming a physical GPIO.

The PCH implements vGPIO39 (in GPIO community 1), which is specifically used for SD card detection as an interrupt generation. If the PCH integrated SD card is utilized, in conjunction of the SD_CD# pin to be used as card detect, a physical GPIO pin is

required for interrupt generation. vGPIO39 is intended to replace the need for this addition physical GPIO if desired. SW needs to program the vGPIO accordingly to enable this functionality.

§ §

18 Intel® Serial I/O Inter-Integrated Circuit (I^2C) Controllers

18.1 Acronyms

Acronyms	Description
I^2C	Inter-Integrated Circuit
PIO	Programmed Input/Output
SCL	Serial Clock Line
SDA	Serial Data Line

18.2 References

Specification	Location
The I^2C Bus Specification, Version 5	www.nxp.com/documents/user_manual/UM10204.pdf?

18.3 Feature Overview

The PCH implements six I^2C controllers for six independent I^2C interfaces, I2C0-I2C5. Each interface is a two-wire serial interface consisting of a serial data line (SDA) and a serial clock (SCL).

I2C4 and I2C5 only implement the I^2C host controllers and do not incorporate a DMA controller. Therefore, I2C4 and I2C5 are restricted to operate in PIO mode only.

The I^2C interfaces support the following features:

- Speed: standard mode (up to 100 Kb/s), fast mode (up to 400 Kb/s), fast mode plus (up to 1 MB/s) and High speed mode (up to 3.2 Mb/s).
- 1.8V or 3.3V support (depending on the voltage supplied to the I^2C signal group)
- Master I^2C operation only
- 7-bit or 10-bit addressing
- 7-bit or 10-bit combined format transfers
- Bulk transmit mode
- Ignoring CBUS addresses (an older ancestor of I^2C used to share the I^2C bus)
- Interrupt or polled-mode operation
- Bit and byte waiting at all bus speed
- Component parameters for configurable software driver support
- Programmable SDA hold time (t_{HD}; DAT)

- DMA support with 64-byte DMA FIFO per channel (up to 32-byte burst)
- 64-byte Tx FIFO and 64-byte Rx FIFO
- SW controlled serial data line (SDA) and serial clock (SCL)

Notes:

1. The controllers must only be programmed to operate in master mode only. I²C slave mode is not supported.
2. I²C multi masters is not supported.
3. Simultaneous configuration of Fast Mode and Fast Mode Plus/High speed mode is not supported.
4. I²C General Call is not supported.

18.4 Signal Description

Name	Type	Description
I2C0_SDA/ GPP_C16	I/OD	**I²C Link 0 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
I2C0_SCL/ GPP_C17	I/OD	**I²C Link 0 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
I2C1_SDA/ GPP_C18	I/OD	**I²C Link 1 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
I2C1_SCL/ GPP_C19	I/OD	**I²C Link 1 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
I2C2_SDA/ GPP_H4	I/OD	**I²C Link 2 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
I2C2_SCL/ GPP_H5	I/OD	**I²C Link 2 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
I2C3_SDA/ GPP_H6	I/OD	**I²C Link 3 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
I2C3_SCL/ GPP_H7	I/OD	**I²C Link 3 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
I2C4_SDA / GPP_H8 / CNV_MUART2_RXD	I/OD	**I²C Link 4 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
I2C4_SCL/ GPP_H9 / CNV_MUART2_TXD	I/OD	**I²C Link 4 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
I2C5_SDA/ GPP_B9 / ISH_I2C2_SDA	I/OD	**I²C Link 5 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
I2C5_SCL/ GPP_B10 / ISH_I2C2_SCL	I/OD	**I²C Link 5 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.

18.5 Integrated Pull-Ups and Pull-Downs

None

18.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S3/S4/S5	Deep Sx
I2C[5:0]_SDA	Primary	Undriven	Undriven	Undriven	OFF
I2C[5:0]_SCL	Primary	Undriven	Undriven	Undriven	OFF

Notes:
1. Reset reference for primary well pins is RSMRST#.

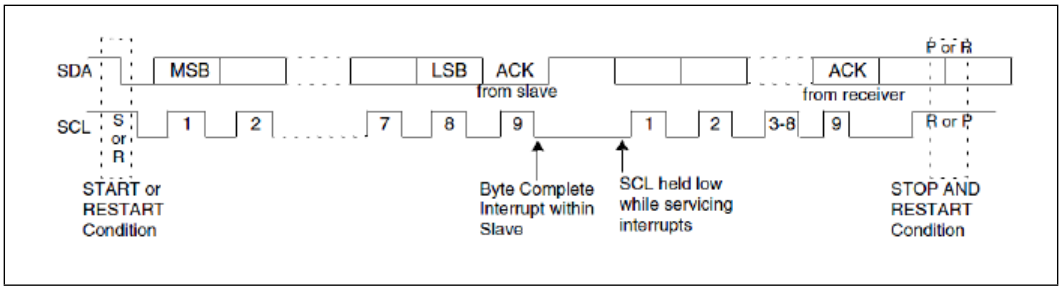
18.7 Functional Description

18.7.1 Protocols Overview

For more information on the I^2C protocols and command formats, refer to the industry I^2C specification. Below is a simplified description of I^2C bus operation:

- The master generates a START condition, signaling all devices on the bus to listen for data.
- The master writes a 7-bit address, followed by a read/write bit to select the target device and to define whether it is a transmitter or a receiver.
- The target device sends an acknowledge bit over the bus. The master must read this bit to determine whether the addressed target device is on the bus.
- Depending on the value of the read/write bit, any number of 8-bit messages can be transmitted or received by the master. These messages are specific to the I^2C device used. After 8 message bits are written to the bus, the transmitter will receive an acknowledge bit. This message and acknowledge transfer continues until the entire message is transmitted.
- The message is terminated by the master with a STOP condition. This frees the bus for the next master to begin communications. When the bus is free, both data and clock lines are high.

Figure 18-1. Data Transfer on the I^2C Bus



18.7.1.1 Combined Formats

The PCH I^2C controllers support mixed read and write combined format transactions in both 7-bit and 10-bit addressing modes.

The PCH controllers do not support mixed address and mixed address format (which means a 7-bit address transaction followed by a 10-bit address transaction or vice versa) combined format transaction.

To initiate combined format transfers, IC_CON.IC_RESTSART_EN should be set to 1. With this value set and operating as a master, when the controller completes an I^2C transfer, it checks the transmit FIFO and executes the next transfer. If the direction of this transfer differs from the previous transfer, the combined format is used to issue the transfer. If the transmit FIFO is empty when the current I^2C transfer completes, a STOP is issued and the next transfer is issued following a START condition.

18.7.2 DMA Controller

The I^2C controllers 0 to 3 (I2C0 - I2C3) each has an integrated DMA controller. The I^2C controller 4 and 5 (I2C4 and I2C5) only implement the I^2C host controllers and do not incorporate a DMA. Therefore, I2C4 and I2C5 are restricted to operate in PIO mode only.

18.7.2.1 DMA Transfer and Setup Modes

The DMA can operate in the following modes:

1. Memory to peripheral transfers. This mode requires the peripheral to control the flow of the data to itself.
2. Peripheral to memory transfer. This mode requires the peripheral to control the flow of the data from itself.

The DMA supports the following modes for programming:

1. Direct programming. Direct register writes to DMA registers to configure and initiate the transfer.
2. Descriptor based linked list. The descriptors will be stored in memory (such as DDR or SRAM). The DMA will be informed with the location information of the descriptor. DMA initiates reads and programs its own register. The descriptors can form a linked list for multiple blocks to be programmed.
3. Scatter Gather mode.

18.7.2.2 Channel Control

- The source transfer width and destination transfer width is programmable. The width can be programmed to 1, 2, or 4 bytes.
- Burst size is configurable per channel for source and destination. The number is a power of 2 and can vary between 1,2,4,...,128. This number times the transaction width gives the number of bytes that will be transferred per burst.
- Individual channel enables. If the channel is not being used, then it should be clock gated.
- Programmable Block size and Packing/Unpacking. Block size of the transfer is programmable in bytes. The block size is not be limited by the source or destination transfer widths.
- Address incrementing modes: The DMA has a configurable mechanism for computing the source and destination addresses for the next transfer within the current block. The DMA supports incrementing addresses and constant addresses.
- Flexibility to configure any hardware handshake sideband interface to any of the DMA channels
- Early termination of a transfer on a particular channel.

18.7.3 Reset

Each host controller has an independent reset associated with it. Control of these resets is accessed through the Reset Register.

Each host controller and DMA will be in reset state once powered ON and require SW (BIOS or driver) to write into specific reset register to bring the controller from reset state into operational mode.

Note: To avoid a potential I^2C peripheral deadlock condition where the reset goes active in the middle of a transaction, the I^2C controller must be idle before a reset can be initiated.

18.7.4 Power Management

18.7.4.1 Device Power Down Support

To power down peripherals connected to PCH I^2C bus, the idle configured state of the I/O signals is retained to avoid voltage transitions on the bus that can affect the connected powered peripheral. Connected devices are allowed to remain in the D0 active or D2 low power states when I^2C bus is powered off (power gated). The PCH HW will prevent any transitions on the serial bus signals during a power gate event.

18.7.4.2 Latency Tolerance Reporting (LTR)

Latency Tolerance Reporting is used to allow the system to optimize internal power states based on dynamic data, comprehending the current platform activity and service latency requirements. The interface supports this by reporting its service latency requirements to the platform power management controller using LTR registers.

The controller's latency tolerance reporting can be managed by one of the two following schemes. The platform integrator must choose the correct scheme for managing latency tolerance reporting based on the platform, OS and usage.

1. Platform/HW Default Control. This scheme is used for usage models in which the controller's state correctly informs the platform of the current latency requirements.
2. Driver Control. This scheme is used for usage models in which the controller state does not inform the platform correctly of the current latency requirements. If the FIFOs of the connected device are much smaller than the controller FIFOs, or the connected device's end to end traffic assumptions are much smaller than the latency to restore the platform from low power state, driver control should be used.

18.7.5 Interrupts

I^2C interface has an interrupt line which is used to notify the driver that service is required.

When an interrupt occurs, the device driver needs to read the host controller, DMA interrupt status and TX completion interrupt registers to identify the interrupt source. Clearing the interrupt is done with the corresponding interrupt register in the host controller or DMA.

All interrupts are active high and their behavior is level triggered.

18.7.6 Error Handling

Errors that might occur on the external I^2C signals are comprehended by the I^2C host controller and reported to the I^2C bus driver through the MMIO registers.

18.7.7 Programmable SDA Hold Time

PCH includes a software programmable register to enable dynamic adjustment of the SDA hold time, if needed.

§ §

19 Gigabit Ethernet Controller

19.1 Acronyms

Acronyms	Description
GbE	Gigabit Ethernet

19.2 References

Specification	Location
Alert Standard Format Specification, Version 1.03	http://www.dmtf.org/standards/asf
IEEE 802.3 Fast Ethernet	http://standards.ieee.org/getieee802/
Intel® Ethernet Connection I219 Datasheet	http://www.intel.com/content/www/us/en/embedded/products/networking/ethernet-connection-i219-datasheet.html

19.3 Overview

The Gigabit Ethernet controller(D31:F6) in conjunction with the Intel® Ethernet Connection I219 provides a complete LAN solution. This chapter describes the behavior of the Gigabit Ethernet Controller. The Gigabit Ethernet Controller can operate at multiple speeds (10/100/1000 Mbps) and in either full duplex or half duplex mode.

19.4 Signal Description

Table 19-1. GbE LAN Signals (Sheet 1 of 2)

Name	Type	Description
PCIE7_TXP **PCIE7_TXN** **PCIE8_TXP** **PCIE8_TXN** **PCIE9_TXP** **PCIE9_TXN** **PCIE13_TXP** **PCIE13_TXN** **PCIE14_TXP** **PCIE14_TXN**	O	Refer to Chapter 22, "PCI Express* (PCIe*)" for details on the PCI Express transmit signals. ***Note:*** For PCH-U, the Intel® Ethernet Connection I219 can be connected to one of the following PCI Express ports 7, 8, 9, 13 or 14.

Table 19-1. GbE LAN Signals (Sheet 2 of 2)

Name	Type	Description
PCIE7_RXP **PCIE7_RXN** **PCIE8_RXP** **PCIE8_RXN** **PCIE9_RXP** **PCIE9_RXN** **PCIE13_RXP** **PCIE13_RXN** **PCIE14_RXP** **PCIE14_RXN**	I	Refer to Chapter 22, "PCI Express* (PCIe*)" for details on the PCI Express transmit signals. ***Note:*** For PCH-U, the Intel® Ethernet Connection I219 can be connected to one of the following PCI Express ports 7, 8, 9, 13 or 14.
SML0DATA/GPP_C4	I/OD	Refer to Chapter 26, "System Management Interface and SMLink" for details on the SML0DATA signal. ***Note:*** The Intel® Ethernet Connection I219 connects to SML0DATA signal.
SML0CLK/GPP_C3	I/OD	Refer to Chapter 26, "System Management Interface and SMLink" for details on the SML0CLK signal. ***Note:*** The Intel® Ethernet Connection I219 connects to SML0CLK signal.
LANPHYPC/GPD11	O	**LAN PHY Power Control:** LANPHYPC should be connected to LAN_DISABLE_N on the PHY. PCH will drive LANPHYPC low to put the PHY into a low power state when functionality is not needed. ***Note:*** LANPHYPC can only be driven low if SLP_LAN# is de-asserted. ***Note:*** Signal can instead be used as GPD11.
SLP_LAN#	O	**LAN Sub-System Sleep Control:** If the Gigabit Ethernet Controller is enabled, when SLP_LAN# is de-asserted it indicates that the PHY device must be powered. When SLP_LAN# is asserted, power can be shut off to the PHY device. ***Note:*** If Gigabit Ethernet Controller is statically disabled via BIOS, SLP_LAN# will be driven low.
LAN_WAKE#/GPD2	I	**LAN WAKE:** LAN Wake Indicator from the GbE PHY. ***Note:*** LAN_WAKE# functionality is only supported with Intel PHY I219. Connection of a third party LAN device's wake signal to LAN_WAKE# is not supported.

19.5 Integrated Pull-Ups and Pull-Downs

Table 19-2. Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value Ω	Notes
LAN_WAKE#/GPD2	External Pull-up required. Internal Pull-down may be enabled in DeepSx	4.7 kohm +/- 5%	Note: 10 kohm +/- 5% pull-up resistor is also acceptable

19.6 I/O Signal Planes and States

Table 19-3. Power Plane and States for Output Signals (Sheet 1 of 2)

Signal Name	Power Plane	During Reset[3]	Immediately after Reset[3]	S3/S4/S5	Deep Sx
LANPHYPC /GPD11	DSW	Undriven	Undriven	Undriven[1]	Undriven[1]
SLP_LAN#	DSW	0/1[2]	0/1[2]	0/1[2]	0/1[2]

Table 19-3. Power Plane and States for Output Signals (Sheet 2 of 2)

Signal Name	Power Plane	During Reset[3]	Immediately after Reset[3]	S3/S4/S5	Deep Sx
Note: 1. Based on wake events and Intel® CSME state. 2. Configurable based on BIOS settings: '0' When LAN controller is configured as "Disabled" in BIOS, SLP_LAN# will drive "Low";'1' When LAN controller is configured as "Enabled" in BIOS, SLP_LAN# will drive "High". 3. Reset reference for DSW well pins is DSW_PWROK.					

Table 19-4. Power Plane and States for Input Signals

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
LAN_WAKE#/ GPD2	DSW	Undriven	Undriven	Undriven	Undriven

19.7 Functional Description

The PCH integrates a Gigabit Ethernet (GbE) controller. The integrated GbE controller is compatible with the Intel® Ethernet Connection I219. The integrated GbE controller provides two interfaces for 10/100/1000 Mbps and manageability operation:

- Data link based on PCI Express* – A high-speed interface that uses PCIe* electrical signaling at half speed and custom logical protocol for active state operation mode.
- System Management Link (SMLink0)—A low speed connection for low power state mode for manageability communication only. The frequency of this connection can be configured to one of three different speeds (100 KHz, 400 KHz or 1 MHz).

The Intel® Ethernet Connection I219 only runs at a speed of 1250 Mbps, which is 1/2 of the 2.5 GB/s PCI Express frequency. Each of the PCI Express* root ports in the PCH have the ability to run at the 1250-Mbps rate. There is no need to implement a mechanism to detect that the Platform LAN Device is connected. The port configuration (if any), attached to the Platform LAN Device, is pre-loaded from the NVM. The selected port adjusts the transmitter to run at the 1250-Mbps rate and does not need to be PCI Express compliant.

Note: PCIe* validation tools cannot be used for electrical validation of this interface—however, PCIe* layout rules apply for on-board routing.

The integrated GbE controller operates at full-duplex at all supported speeds or half-duplex at 10/100 Mbps. It also adheres to the *IEEE 802.3x Flow Control Specification.*

Note: GbE operation (1000 Mbps) is only supported in S0 mode. In Sx modes, the platform LAN Device may maintain 10/100 Mbps connectivity and use the SMLink interface to communicate with the PCH.

The integrated GbE controller provides a system interface using a PCI Express function. A full memory-mapped or I/O-mapped interface is provided to the software, along with DMA mechanisms for high performance data transfer.

The integrated GbE controller features are:

- Network Features
  - — Compliant with the 1 GB/s Ethernet 802.3, 802.3u, 802.3ab specifications
  - — Multi-speed operation: 10/100/1000 Mbps

  - — Full-duplex operation at 10/100/1000 Mbps: Half-duplex at 10/100 Mbps
  - — Flow control support compliant with the 802.3X specification
  - — VLAN support compliant with the 802.3q specification
  - — MAC address filters: perfect match unicast filters; multicast hash filtering, broadcast filter and promiscuous mode
  - — PCI Express/SMLink interface to GbE PHYs
- Host Interface Features
  - — 64-bit address master support for systems using more than 4 GB of physical memory
  - — Programmable host memory receive buffers (256 bytes to 16 KB)
  - — Intelligent interrupt generation features to enhance driver performance
  - — Descriptor ring management hardware for transmit and receive
  - — Software controlled reset (resets everything except the configuration space)
  - — Message Signaled Interrupts
- Performance Features
  - — Configurable receive and transmit data FIFO, programmable in 1 KB increments
  - — TCP segmentation off loading features
  - — Fragmented UDP checksum off load for packet reassembly
  - — IPv4 and IPv6 checksum off load support (receive, transmit, and large send)
  - — Split header support to eliminate payload copy from user space to host space
  - — Receive Side Scaling (RSS) with two hardware receive queues
  - — Supports 9018 bytes of jumbo packets
  - — Packet buffer size 32 KB
  - — TimeSync off load compliant with 802.1as specification
  - — Platform time synchronization
- Power Management Features
  - — 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, ACPI)
  - — MAC power down at Sx, DM-Off with and without WoL
  - — Auto connect battery saver at S0 no link and Sx no link
  - — Energy Efficient Ethernet (EEE) support
  - — Latency Tolerance Reporting (LTR)
  - — ARP and ND proxy support through LAN Connected Device proxy
  - — Wake on LAN (WoL) from Deep Sx
  - — Windows* InstantGo* Support

19.7.1 GbE PCI Express* Bus Interface

The GbE controller has a PCI Express interface to the host processor and host memory. The following sections detail the bus transactions.

19.7.1.1 Transaction Layer

The upper layer of the host architecture is the transaction layer. The transaction layer connects to the device GbE controller using an implementation specific protocol. Through this GbE controller-to-transaction-layer protocol, the application-specific parts of the device interact with the subsystem and transmit and receive requests to or from the remote agent, respectively.

19.7.1.2 Data Alignment

19.7.1.2.1 4-KB Boundary

PCI requests must never specify an address/length combination that causes a memory space access to cross a 4-KB boundary. It is hardware's responsibility to break requests into 4-KB aligned requests (if needed). This does not pose any requirement on software. However, if software allocates a buffer across a 4-KB boundary, hardware issues multiple requests for the buffer. Software should consider aligning buffers to a 4-KB boundary in cases where it improves performance. The alignment to the 4-KB boundaries is done by the GbE controller. The transaction layer does not do any alignment according to these boundaries.

19.7.1.2.2 PCI Request Size

PCI requests are 128 bytes or less and are aligned to make better use of memory controller resources. Writes, however, can be on any boundary and can cross a 64-byte alignment boundary.

19.7.1.3 Configuration Request Retry Status

The integrated GbE controller might have a delay in initialization due to an NVM read. If the NVM configuration read operation is not completed and the device receives a configuration request, the device responds with a configuration request retry completion status to terminate the request, and thus effectively stalls the configuration request until such time that the sub-system has completed local initialization and is ready to communicate with the host.

19.7.2 Error Events and Error Reporting

19.7.2.1 Completer Abort Error Handling

A received request that violates the LAN Controller programing model will be discarded, for non posted transactions an unsuccessful completion with CA completion status will be returned. For posted transactions if both SERR# enable and URRE# enable are enabled, the LAN Controller will assert SERR#.

19.7.2.2 Unsupported Request Error Handling

A received unsupported request to the LAN Controller will be discarded, for non posted transactions an unsuccessful completion with UR completion status will be returned. The URD bit will be set in ECTL register, If both SERR# enable and URRE# enable are enabled, the LAN Controller will assert SERR#. For posted transactions, if both SERR# enable and URRE# enable are enabled, the LAN Controller will assert SERR#.

19.7.3 Ethernet Interface

The integrated GbE controller provides a complete CSMA/CD function supporting IEEE 802.3 (10 Mbps), 802.3u (100 Mbps) implementations. It also supports the IEEE 802.3z and 802.3ab (1000 Mbps) implementations. The device performs all of the functions required for transmission, reception, and collision handling called out in the standards.

The mode used to communicate between the PCH and the Intel® Ethernet Connection I219 supports 10/100/1000 Mbps operation, with both half- and full-duplex operation at 10/100 Mbps, and full-duplex operation at 1000 Mbps.

19.7.3.1 Intel® Ethernet Connection I219

The integrated GbE controller and the Intel® Ethernet Connection I219 communicate through the PCIe* and SMLink0 interfaces. All integrated GbE controller configuration is performed using device control registers mapped into system memory or I/O space. The Platform LAN Phy is configured using the PCI Express or SMLink0 interface.

The integrated GbE controller supports various modes as listed in Table 19-5.

Table 19-5. LAN Mode Support

Mode	System State	Interface Active	Connections
Normal 10/100/1000 Mbps	S0	PCI Express	Intel® Ethernet Connection I219
Manageability and Wake-on-LAN	Sx	SMLink0	Intel® Ethernet Connection I219
Wake-on-LAN	Deep Sx	LAN_WAKE#	Intel® Ethernet Connection I219

19.7.4 PCI Power Management

The integrated GbE controller supports the Advanced Configuration and Power Interface (ACPI) specification as well as Advanced Power Management (APM). This enables the network-related activity (using an internal host wake signal) to wake up the host. For example, from Sx (S3–S5) and Deep Sx to S0.

Note: The Intel® Ethernet Connection I219 must be powered during the Deep Sx state in order to support host wake up from Deep Sx. GPD_2_LAN_WAKE# on the PCH must be configured to support wake from Deep Sx and must be connected to LANWAKE_N on the Platform LAN Connect Device. The SLP_LAN# signal must be driven high (de-asserted) in the Deep Sx state to maintain power to the Platform LAN Connect Device.

The integrated GbE controller contains power management registers for PCI and supports D0 and D3 states. PCIe* transactions are only allowed in the D0 state, except for host accesses to the integrated GbE controller's PCI configuration registers.

Note: SLP_LAN# pin behavior are detailed in Section 23.7.8.7, "SLP_LAN# Pin Behavior"

§ §

20 Integrated Sensor Hub (ISH)

20.1 Acronyms

Acronyms	Description
Intel® CSME	Intel® Converged Security and Management Engine
I^2C	Inter-Integrated Circuit
IPC	Inter Process Communication
SPI	Serial Peripheral Interface
ISH	Integrated Sensor Hub
PMU	Power Management Unit
SRAM	Static Random Access Memory
UART	Universal Asynchronous Receiver/Transmitter

20.2 References

Specification	Location
I^2C Specification Version 6.0	http://www.nxp.com/docs/en/user-guide/UM10204.pdf

20.3 Feature Overview

The Integrated Sensor Hub (ISH) serves as the connection point for many of the sensors on a platform. The ISH is designed with the goal of "Always On, Always Sensing" and it provides the following functions to support this goal:

- Acquisition/sampling of sensor data.
- The ability to combine data from individual sensors to create a more complex virtual sensor that can be directly used by the firmware/OS.
- Low power operation through clock and power gating of the ISH blocks together with the ability to manage the power state of the external sensors.
- The ability to operate independently when the host platform is in a low power state (S0ix only).
- Ability to provide sensor-related data to other subsystems within the PCH, such as the Intel® CSME.

The ISH consists of the following key components:

- A combined cache for instructions and data.
  - — ROM space intended for the bootloader.
  - — SRAM space for code and data.
- Interfaces to sensor peripherals (I^2C, UART, SPI, GPIO).
- An interface to main memory.
- Out of Band signals for clock and wake-up control.
- Inter Process Communications to the Host and Intel® CSME.

- Part of the PCI tree on the host.

20.3.1 ISH I^{2}C Controllers

The ISH supports three I^{2}C controllers capable of operating at speeds up to 2.4 Mbps each. The I^{2}C controllers are completely independent of each other: they do not share any pins, memory spaces, or interrupts.

The ISH's I^{2}C host controllers share the same general specifications:

- Master Mode Only (all peripherals must be slave devices)
- Support for the following operating speeds:
  - — Standard mode: 100 kbps
  - — Fast Mode: 400 kbps
  - — Fast Mode Plus: 1 000 kbps
  - — High Speed Mode: 2400 kbps
- Support for both 7-bit and 10-bit addressing formats on the I^{2}C bus
- FIFO of 64 bytes with programmable watermarks/thresholds

20.3.2 ISH UART Controller

The ISH has two UART ports, each comprised of a four-wire, bi-directional point-to-point connection between the ISH and a peripheral.

The UART has the following Capabilities:

- Support for operating speeds up to 4 Mbps
- Support for auto flow control using the RTS#/CTS# signals
- 64-byte FIFO
- DMA support to allow direct transfer to the ISH local SRAM without intervention by the controller. This saves interrupts on packets that are longer than the FIFO or when there are back-to-back packets to send or receive.

20.3.3 ISH GSPI Controller

The ISH supports one SPI controller comprises of four-wired interface connecting the ISH to external sensor devices.

The SPI controller includes:

- Master Mode Only
- Single Chip Select
- Half Duplex operation only
- Programmable SPI clock frequency range with maximum rate of 24 Mbits/sec
- FIFO of 64 bytes with programmable thresholds
- Support Programmable character length (2 to 16 bits)

20.3.4 ISH GPIOs

- The ISH supports eight dedicated GPIOs.

20.4 Signal Description

Name	Type	Description
ISH_I2C0_SDA/GPP_B5	I/OD	**ISH I²C 0 Data**
ISH_I2C0_SCL/GPP_B6	I/OD	**ISH I²C 0 Clk**
ISH_I2C1_SDA/GPP_B7	I/OD	**ISH I²C 1 Data**
ISH_I2C1_SCL/GPP_B8	I/OD	**ISH I²C 1 Clk**
ISH_I2C2_SDA / GPP_B9 / I2C5_SDA	I/OD	**ISH I²C 2 Data**
ISH_I2C2_SCL / GPP_B10 / I2C5_SCL	I/OD	**ISH I²C 2 Clk**
ISH_GP0/GPP_D0	I/O	**ISH GPIO 0**
ISH_GP1/GPP_D1	I/O	**ISH GPIO 1**
ISH_GP2/GPP_D2	I/O	**ISH GPIO 2**
ISH_GP3/GPP_D3	I/O	**ISH GPIO 3**
ISH_GP4/GPP_D17	I/O	**ISH GPIO 4**
ISH_GP5/GPP_D18	I/O	**ISH GPIO 5**
ISH_GP6 / GPP_E15	I/O	**ISH GPIO 6**
ISH_GP7/ GPP_E16	I/O	**ISH GPIO 7**
ISH_UART0_TXD / GPP_D14	O	**ISH UART 0 Transmit Data**
ISH_UART0_RXD /GPP_D13	I	**ISH UART 0 Receive Data**
ISH_UART0_RTS# /GPP_D15 / GSPI2_CS1# / IMGCLKOUT5	O	**ISH UART 0 Request To Send**
ISH_UART0_CTS# /GPP_D16 / CNV_WCEN	I	**ISH UART 0 Clear to Send**
ISH_UART1_TXD /GPP_C13 / UART1_TXD	O	**ISH UART 1 Transmit Data**
ISH_UART1_RXD /GPP_C12 / UART1_RXD	I	**ISH UART 1 Receive Data**
ISH_UART1_RTS# /GPP_C14 / UART1_RTS#	O	**ISH UART 1 Request To Send**
ISH_UART1_CTS# / GPP_C15 / UART1_CTS#	I	**ISH UART 1 Clear to Send**
ISH_SPI_CS# / GPP_D9 / DDP3_CTRLCLK / GSPI2_CS0# / TBT_LSX2_TXD	O	**ISH Generic SPI 2 Chip Select**
ISH_SPI_CLK / GPP_D10 / DDP3_CTRLDATA / GSPI2_CLK / TBT_LSX2_RXD	O	**ISH Generic SPI 2 Clock**
ISH_SPI_MISO / GPP_D11 / DDP4_CTRLCLK / GSPI2_MISO / TBT_LSX3_TXD	I	**ISH Generic SPI 2 MISO**
ISH_SPI_MOSI / GPP_D12 / DDP4_CTRLDATA / GSPI2_-MOSI / TBT_LSX3_RXD	O	**ISH Generic SPI 2 MOSI**

20.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
ISH_SPI_MISO	Pull-Down	20k ± 30%	

20.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S3/S4/S5	Deep Sx
ISH_I2C0_SDA	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C0_SCL	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C1_SDA	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C1_SCL	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C2_SDA	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C2_SCL	Primary	Undriven	Undriven	Undriven	OFF
ISH_GP[7:0]	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART0_TXD	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART0_RXD	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART0_RTS#	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART0_CTS#	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART1_TXD	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART1_RXD	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART1_RTS#	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART1_CTS#	Primary	Undriven	Undriven	Undriven	OFF
ISH_SPI_CS#	Primary	Undriven	Undriven	Undriven	OFF
ISH_SPI_CLK	Primary	Undriven	Undriven	Undriven	OFF
ISH_SPI_MISO	Primary	Undriven	Undriven	Undriven	OFF
ISH_SPI_MOSI	Primary	Undriven	Undriven	Undriven	OFF

Notes:
1. Reset reference for primary well pins is RSMRST#.

20.7 Functional Description

20.7.1 ISH Micro-Controller

The ISH is operated by a micro-controller. This core provides localized sensor aggregation and data processing, thus off loading the processor and lowering overall platform average power. The core supports an in-built local APIC that receives messages from the IOAPIC. A local boot ROM with FW for initialization is also part of the core.

20.7.2 SRAM

The local SRAM is used for ISH FW code storage and to read/write operational data. The local SRAM block includes both the physical SRAM as well as the controller logic. The SRAM is a total of 640K bytes organized into banks of 32 kB each and is 32-bit wide. The SRAM is shared with Intel® CSME as shareable memory. To protect against memory errors, the SRAM includes ECC support. The ECC mechanism is able to detect multi-bit errors and correct for single bit errors. The ISH firmware has the ability to put unused SRAM banks into lower power states to reduce power consumption.

20.7.3 PCI Host Interface

The ISH provides access to PCI configuration space via a PCI Bridge. Type 0 Configuration Cycles from the host are directed to the PCI configuration space.

20.7.3.1 MMIO Space

A memory-mapped Base Address Register (BAR0) with a set of functional memory-mapped registers is accessible to the host via the Bridge. These registers are owned by the driver running on the Host OS.

The bridge also supports a second BAR (BAR1) that is an alias of the PCI Con figuration space. It is used only in ACPI mode (that is, when the PCI con figuration space is hidden).

20.7.3.2 DMA Controller

The DMA controller supports up to 64-bit addressing.

20.7.3.3 PCI Interrupts

The PCI bridge supports standard PCI interrupts, delivered using IRQx to the system IOAPIC and not using an MSI to the host CPU.

20.7.3.4 PCI Power Management

PME is not supported in ISH.

20.7.4 Power Domains and Management

20.7.4.1 ISH Power Management

The various functional blocks within the ISH are all on the primary power plane within the PCH. The ISH is only intended for use during S0 and S0ix states. There is no support for operation in S3, S4, or S5 states. Thus, the system designer must ensure that the inputs to the ISH signals are not driven high while the PCH is in S3–S5 state.

The unused banks of the ISH SRAM can be power-gated by the ISH Firmware.

20.7.4.2 External Sensor Power Management

External sensors can generally be put into a low power state through commands issued over the I/O interface (I^2C). Refer to the datasheets of the individual sensors to obtain the commands to be sent to the peripheral.

20.7.5 ISH IPC

The ISH has IPC channels for communication with the Host Processor and Intel® CSME. The functions supported by the ISH IPC block are listed below.

Function 1: Allows for messages and interrupts to be sent from an initiator (such as the ISH) and a target (such as the Intel® CSME). The supported initiator -> target flows using this mechanism are shown in the table below

Table 20-1. IPC Initiator -> Target flows

Initiator	Target
ISH	Host processor
Host processor	ISH
ISH	Intel® CSME
Intel® CSME	ISH

Function 2: Provides status registers and remap registers that assist in the boot flow and debug. These are simple registers with dual access read/write support and cause no interrupts.

20.7.6 ISH Interrupt Handling via IOAPIC (Interrupt Controller)

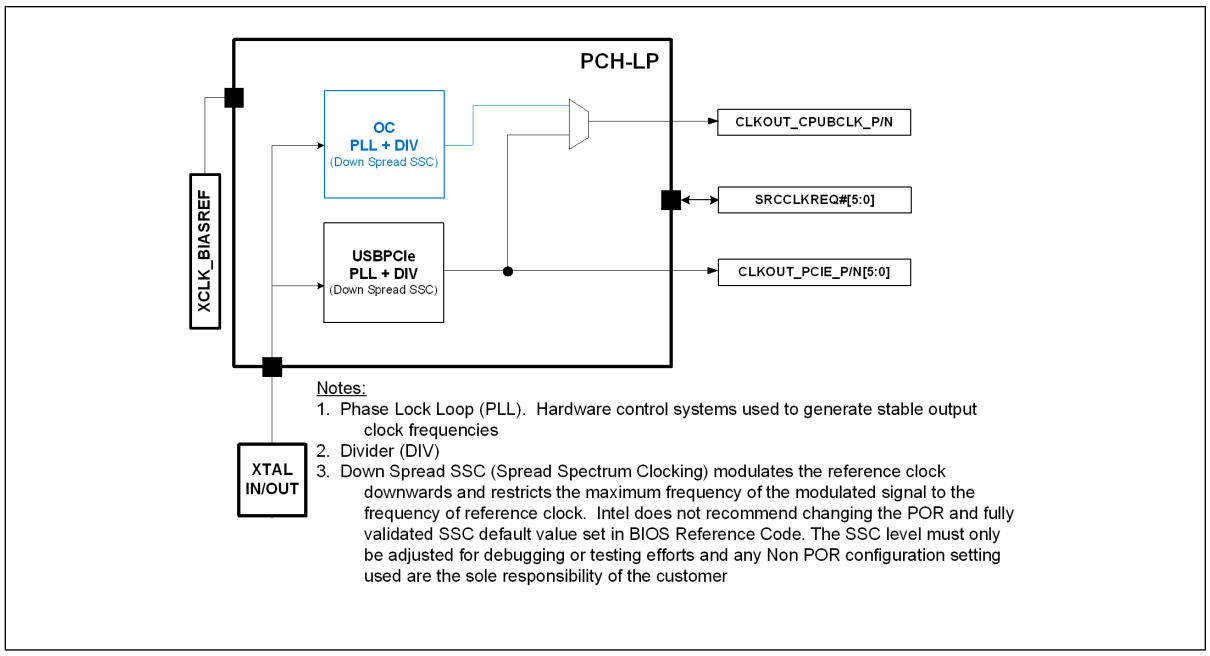
The PCH legacy IOAPIC is the interrupt controller for the ISH. It collects inputs from various internal blocks and sends interrupt messages to the ISH controller. When there is a change on one of its inputs, the IOAPIC sends an interrupt message to the ISH controller.

The PCH IOAPIC allows each interrupt input to be active high or active low and edge or level triggered.

§ §

21 PCH and System Clocks

21.1 Integrated Clock Controller (ICC)



21.2 PCH ICC Clocking Profiles

The PCH ICC hardware includes the following clocking profiles:

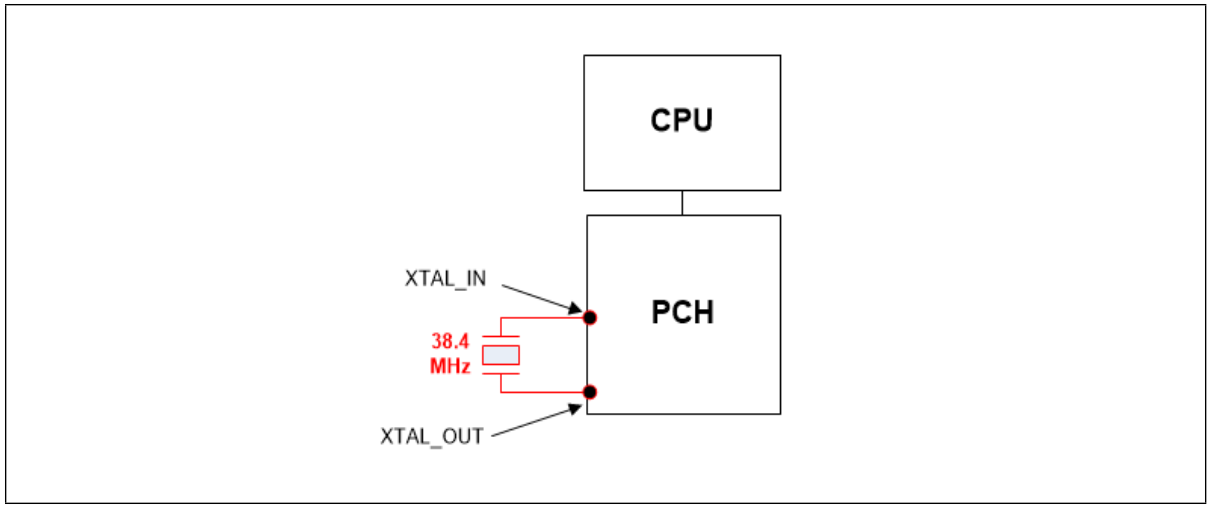
— "Standard" Profile
- OC PLL = Disabled
- USBPCIe PLL = Enabled with Down Spread Spectrum Clocking (SSC)

— "Adaptive" Profile
- OC PLL = Enabled with Down Spread Spectrum Clocking (SSC) and Under Clocking Capability
- USBPCIe PLL = Enabled with Down Spread Spectrum Clocking (SSC)

Note: These PCH ICC Clocking Profiles can be enabled through the Intel® Flash Image Tool. The Standard ICC Profile is set by default and is the recommended ICC Clocking Profile

21.3 PCH ICC XTAL Input Configuration

The PCH ICC supports the following XTAL Input Configuration.



21.4 Signal Descriptions

Name	Type	SSC Capable	Description
PCH U: • CLKOUT_PCIE_P[5:0] • CLKOUT_PCIE_N[5:0] PCH Y: • CLKOUT_PCIE_P[5:1] • CLKOUT_PCIE_N[5:1]	O	Yes	**PCI Express* Clock Output:** Serial Reference 100 MHz PCIe* 3.0 specification compliant differential output clocks to PCIe* devices
PCH U: • SRCCLKREQ#[5:0] PCH Y: • SRCCLKREQ#[5:1]	I/O		**Clock Request:** Serial Reference Clock request signals for PCIe* 100 MHz differential clocks
XTAL_IN	I		**Crystal Input:** Input connection for 38.4 MHz crystal to PCH
XTAL_OUT	O		**Crystal Output:** Output connection for 38.4 MHz crystal to PCH
XCLK_BIASREF	I/O		**Differential Clock Bias Reference:** Used to set BIAS reference for differential clocks

Notes:
1. SSC = Spread Spectrum Clocking. Intel does not recommend changing the Plan of Record and fully validated SSC default value set in BIOS Reference Code. The SSC level must only be adjusted for debugging or testing efforts and any Non POR configuration setting used are the sole responsibility of the customer.
2. The SRCCLKREQ# signals can be configured to map to any of the PCH PCI Express* Root Ports while using any of the CLKOUT_PCIE_P/N differential pairs.

21.5 I/O Signal Pin States

Signal Name	S3/S4/S5	S0 Entry	S0	Deep Sx
CLKOUT_PCIE_P[5:0] CLKOUT_PCIE_N[5:0]	OFF (Gated Low)	Bringing up the Clock	Toggling	OFF (Gated Low)
SRCCLKREQ#[5:0]	Un-driven	Un-driven	Driven	Off

§ §

22 PCI Express* (PCIe*)

22.1 Acronyms

Acronyms	Description
PCIe*	PCI Express* (Peripheral Component Interconnect Express*)

22.2 Signal Description

PCH	Name	Type	Description
U PCH-LP	PCIE[16:1]_TXP PCIE[16:1]_TXN	O	**PCI Express* Differential Transmit Pairs** These are PCI Express* based outbound high-speed differential signals
	PCIE[16:1]_RXP PCIE[16:1]_RXN	I	**PCI Express* Differential Receive Pairs** These are PCI Express* based inbound high-speed differential signals
	PCIE_RCOMPP PCIE_RCOMPN	I	**Impedance Compensation Inputs**
Y PCH-LP	PCIE[14:1]_TXP PCIE[14:1]_TXN	O	**PCI Express* Differential Transmit Pairs** These are PCI Express* based outbound high-speed differential signals
	PCIE[14:1]_RXP PCIE[14:1]_RXN	I	**PCI Express* Differential Receive Pairs** These are PCI Express* based inbound high-speed differential signals
	PCIE_RCOMPP PCIE_RCOMPN	I	**Impedance Compensation Inputs**

22.3 I/O Signal Planes and States

Table 22-1. Power Plane and States for PCI Express* Signals

Signal Name	Type	Power Plane	During Reset[1]	Immediately After Reset[1]	S3/S4/S5	Deep Sx
PCIE_TXP PCIE_TXN	O	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
PCIE_RXP PCIE_RXN	I	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
PCIE_RCOMPP PCIE_RCOMPN	I	Primary	Undriven	Undriven	Undriven	OFF

Notes:
1. Reset reference for primary well pins is RSMRST#.
2. PCIE_RXP\RXN pins transition from un-driven to Internal Pull-down during Reset.

22.4 PCI Express* Port Support Feature Details

Table 22-2. PCI Express* Port Feature Details

PCH	Max Transfer Rate	Max Device (Ports)	Max Lanes	PCIe* Gen Type	Encoding	Transfer Rate (MT/s)	Theoretical Max Bandwidth (GB/s)		
							x1	x2	x4
U PCH-LP	8 GT/s (Gen3)	6	16	1	8b/10b	2500	0.25	0.50	1.00
				2	8b/10b	5000	0.50	1.00	2.00
				3	128b/130b	8000	1.00	2.00	3.94
Y PCH-LP	8 GT/s (Gen3)	5	14	1	8b/10b	2500	0.25	0.50	1.00
				2	8b/10b	5000	0.50	1.00	2.00
				3	128b/130b	8000	1.00	2.00	3.94

Notes:
1. Theoretical Maximum Bandwidth (GB/s) = ((Transfer Rate * Encoding * # PCIe Lanes) /8)/1000
   — Gen3 Example: = ((8000 * 128/130* 4)/8)/1000 = 3.94 GB/s
2. When GbE is enabled on a PCIe Root Port, the Max. Device (Ports) value listed is reduced by a factor of 1

Figure 22-1. Supported PCI Express* Link Configurations

PCH-LP		PCIe* Controller #1				PCIe* Controller #2				PCIe* Controller #3 Cycle Router #2				PCIe* Controller #4 Cycle Router #3			
Flex I/O Lanes		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PCIe* Lanes		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Link Lanes	1x4	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
	1x4 LR	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0
	2x2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	2x2 LR	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
	1x2+2x1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
	2x1+1x2	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
	4x1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Base-U	1x4					RP5				RP9				RP13			
	1x4 LR					RP5				RP9				RP13			
	2x2					RP5		RP7		RP9		RP11		RP13		RP15	
	2x2 LR					RP7		RP5		RP11		RP9		RP15		RP13	
	1x2+2x1					RP5		RP7	RP8	RP9		RP11	RP12	RP13		RP15	RP16
	2x1+1x2					RP8	RP7	RP5		RP12	RP11	RP9		RP16	RP15	RP13	
	4x1					RP5	RP6	RP7	RP8	RP9	RP10	RP11	RP12	RP13	RP14	RP15	RP16
Premium-U	1x4	RP1				RP5				RP9				RP13			
	1x4 LR	RP1				RP5				RP9				RP13			
	2x2	RP1		RP3		RP5		RP7		RP9		RP11		RP13		RP15	
	2x2 LR	RP3		RP1		RP7		RP5		RP11		RP9		RP15		RP13	
	1x2+2x1	RP1		RP3	RP4	RP5		RP7	RP8	RP9		RP11	RP12	RP13		RP15	RP16
	2x1+1x2	RP4	RP3	RP1		RP8	RP7	RP5		RP12	RP11	RP9		RP16	RP15	RP13	
	4x1	RP1	RP2	RP3	RP4	RP5	RP6	RP7	RP8	RP9	RP10	RP11	RP12	RP13	RP14	RP15	RP16
Premium-Y	1x4	RP1				RP5				RP9				RP13			
	1x4 LR	RP1				RP5				RP9							
	2x2	RP1		RP3		RP5		RP7		RP9		RP11		RP13			
	2x2 LR	RP3		RP1		RP7		RP5		RP11		RP9					
	1x2+2x1	RP1		RP3	RP4	RP5		RP7	RP8	RP9		RP11	RP12	RP13			
	2x1+1x2	RP4	RP3	RP1		RP8	RP7	RP5		RP12	RP11	RP9					
	4x1	RP1	RP2	RP3	RP4	RP5	RP6	RP7	RP8	RP9	RP10	RP11	RP12	RP13	RP14		

Notes:
1. RP# refers to a specific PCH PCI Express* Root Port #; for example RP3 = PCH PCI Express* Root Port 3

2. A PCIe* Lane is composed of a single pair of Transmit (TX) and Receive (RX) differential pairs, for a total of four data wires per PCIe* Lane (such as, PCIE[3]_TXP/ PCIE[3]_TXN and PCIE[3]_RXP/ PCIE[3]_RXN make up PCIe Lane 3). A connection between two PCIe* devices is known as a PCIe* Link, and is built up from a collection of one or more PCIe* Lanes which make up the width of the link (such as bundling 2 PCIe* Lanes together would make a x2 PCIe* Link). A PCIe* Link is addressed by the lowest number PCIe* Lane it connects to and is known as the PCIe* Root Port (such as a x2 PCIe* Link connected to PCIe* Lanes 3 and 4 would be called x2 PCIe* Root Port 3).
3. The PCIe* Lanes can be configured independently from one another but the max number of configured Root Ports (Devices) must not be exceeded
   - U PCH-LP: A maximum of 6 PCIe* Root Ports (or devices) can be enabled
   - Y PCH-LP: A maximum of 5 PCIe* Root Ports (or devices) can be enabled
   - When a GbE Port is enabled, the maximum number of PCIe* Ports (or devices) that can be enabled reduces based off the following:
     - U PCH-LP: Max PCIe* Ports (or devices) = 6 - GbE (0 or 1)
     - Y PCH-LP: Max PCIe* Ports (or devices) = 5 - GbE (0 or 1)
4. Unidentified lanes within a PCIe* Link Configuration are disabled but their physical lanes are used for the identified Root Port
5. Supports up to Two Remapped (Intel® Rapid Storage Technology) PCIe* Storage Devices
   - Cells highlighted in Green identify controllers, configurations, and lanes that can be used for a x2 or x4 Intel® Rapid Storage Technology Remapped PCIe* Storage Device or Intel® Optane™ Memory
6. The PCH PCIe* Root Ports can be configured to map to any of the SRCCLKREQ# PCIe* clock request signals and the CLKOUT_PCIE_P/N PCIe* differential clock signal pairs covered in the “Platform Clocks Design Guidelines” Chapter
7. Reference and understand the PCIe* High Speed I/O Multiplexing details covered in the “Flexible I/O” Chapter
8. Lane Reversal Supported Motherboard PCIe* Configurations = 1x4, 2x1+1x2, and 2x2
   - The 2x1+1x2 configuration is enabled by setting the PCIe* Controller soft straps to 1x2+2x1 with Lane Reversal Enabled
   - 1x4 = 1x4 with Lane Reversal Disabled, 1x4 LR = 1x4 with Lane Reversal Enabled
   - 2x2 = 2x2 with Lane Reversal Disabled, 2x2 LR = 2x2 with Lane Reversal Enabled
9. For unused SATA/PCIe* Combo Lanes, Flex I/O Lanes that can be configured as PCIe* or SATA, the lanes must be statically assigned to SATA or PCIe* via the SATA/PCIe Combo Port Soft Straps. These unused SATA/PCIe* Combo Lanes must not be assigned as polarity based.
   - Refer to the “Flexible I/O” chapter for SATA/PCIe* Combo Lane identification

22.5 Overview/Functional Description

- Interrupt Generation
- PCI Express* Power Management
- Latency Tolerance Reporting (LTR)
- Dynamic Link Throttling
- Port 8xh Decode
- PCI Express* Separate Reference Clock with Independent Spread Spectrum Clocking (SRIS)
- Advanced Error Reporting
- Single Root I/O Virtualization (SR- IOV) Capability with Access Control Services (ACS) and Alternative Routing ID (ARI)
- SERR# Generation
- PCI Express* ExpressCard 1.0 module based hot-plug
- PCI Express* TX and RX Lane Polarity Inversion
- End-to-End PCI Express* Controller Lane Reversal
- Dynamic Link Width Negotiation as a Target
- Dynamic Speed Change
- 256B Maximum Data Payload Size
- PCIe* Subtractive Decode is not supported

  - PCI can still be supported via a PCIe*-to-PCI bridge. However, legacy PCI devices (such as PCMCIA or non-plug-and-play device) that need subtractive decode are not supported.
- Common RefClk RX Architecture support
- Precision Time Measurement (PTM)

22.5.1 Interrupt Generation

The root port generates interrupts on behalf of hot-plug, power management, link bandwidth management, Link Equalization Request and link error events, when enabled. These interrupts can either be pin-based, or can be Message Signal Interrupt (MSI), when enabled.

When an interrupt is generated using the legacy pin, the pin is internally routed to the SoC interrupt controllers. The pin that is driven is based upon the setting of the STRPFUSECFG.PXIP configuration registers.

Table 22-3 summarizes interrupt behavior for MSI and wire-modes. In the table "bits" refers to the hot-plug and PME interrupt bits.

Table 22-3. MSI Versus PCI IRQ Actions

Interrupt Register	Wire-Mode Action	MSI Action
All bits 0	Wire inactive	No action
One or more bits set to 1	Wire active	Send message
One or more bits set to 1, new bit gets set to 1	Wire active	Send message
One or more bits set to 1, software clears some (but not all) bits	Wire active	Send message
One or more bits set to 1, software clears all bits	Wire inactive	No action
Software clears one or more bits, and one or more bits are set on the same clock	Wire active	Send message

22.5.2 PCI Express* Power Management

22.5.2.1 S4/S5 Support

Software initiates the transition to S4/S5 by performing an I/O write to the Power Management Control register in the SoC. After the I/O write completion has been returned to the processor, the Power Management Controller will signal each root port to send a PME_Turn_Off message on the downstream link. The device attached to the link will eventually respond with a PME_TO_Ack followed by sending a PM_Enter_L23 DLLP (Data Link Layer Packet) request to enter L23. The Express ports and Power Management Controller take no action upon receiving a PME_TO_Ack. When all the Express port links are in state L23, the Power Management Controller will proceed with the entry into S4/S5.

22.5.2.2 Device Initiated PM_PME Message

When the system has returned to a working state from a previous low power state, a device requesting service will send a PM_PME message continuously, until acknowledged by the root port. The root port will take different actions depending upon whether this is the first PM_PME that has been received, or whether a previous message has been received but not yet serviced by the operating system.

If this is the first message received (RSTS.PS), the root port will set RSTS.PS, and log the PME Requester ID into RSTS.RID. If an interrupt is enabled using RCTL.PIE, an interrupt will be generated. This interrupt can be either a pin or an MSI if MSI is enabled using MC.MSIE.

If this is a subsequent message received (RSTS.PS is already set), the root port will set RSTS.PP. No other action will be taken.

When the first PME event is cleared by software clearing RSTS.PS, the root port will set RSTS.PS, clear RSTS.PP, and move the requester ID into RSTS.RID.

If RCTL.PIE is set, an interrupt will be generated. If RCTL.PIE is not set, a message will be sent to the power management controller so that a GPE can be set. If messages have been logged (RSTS.PS is set), and RCTL.PIE is later written from a 0b to a 1b, an interrupt will be generated. This last condition handles the case where the message was received prior to the operating system re-enabling interrupts after resuming from a low power state.

22.5.2.3 SMI/SCI Generation

Interrupts for power management events are not supported on legacy operating systems. To support power management on non-PCI Express aware operating systems, PM events can be routed to generate SCI. To generate SCI, MPC.PMCE must be set. When set, a power management event will cause SMSCS.PMCS to be set.

Additionally, BIOS workarounds for power management can be supported by setting MPC.PMME. When this bit is set, power management events will set SMSCS.PMMS, and SMI# will be generated. This bit will be set regardless of whether interrupts or SCI is enabled. The SMI# may occur concurrently with an interrupt or SCI.

22.5.2.4 Latency Tolerance Reporting (LTR)

The root port supports the extended Latency Tolerance Reporting (LTR) capability. LTR provides a means for device endpoints to dynamically report their service latency requirements for memory access to the root port. Endpoint devices should transmit a new LTR message to the root port each time its latency tolerance changes (and initially during boot). The PCH uses the information to make better power management decisions. The processor uses the worst case tolerance value communicated by the PCH to optimize C-state transitions. This results in better platform power management without impacting endpoint functionality.

Note: Endpoint devices that support LTR must implement the reporting and enable mechanism detailed in the PCI-SIG "Latency Tolerance Reporting Engineering Change Notice" (www.pcisig.com).

22.5.3 Dynamic Link Throttling

Root Port supports dynamic link throttling as a mechanism to help lower the overall component power, ensuring that the component never operates beyond the thermal limit of the package. Dynamic link throttling is also used as a mechanism for ensuring that the ICC_{max} current rating of the voltage regulator is never exceeded. The target response time for this particular usage model is < 100 µs.

If dynamic link throttling is enabled, the link will be induced by the Root Port to enter TxL0s and RxL0s based on the throttle severity indication received. To induce the link into TxL0s, new TLP requests and opportunistic flow control update will be blocked. Eventually, in the absence of TLP and DLLP requests, the transmitter side of the link will enter TxL0s.

The periodic flow control update, as required by the PCI Express Base Specification is not blocked. However, the flow control credit values advertised to the component on the other side of the link will not be incremented, even if the periodic flow control update packet is sent. Once the other component runs out of credits, it will eventually enter TxL0s, resulting in the local receiver entering RxL0s.

Each of the Root Ports receives four throttle severity indications; T0, T1, T2, and T3. The throttling response for each of the four throttle severity levels can be independently configured in the Root Port TNPT.TSLxM register fields. This allows the duty cycle of the Throttling Window to be varied based on the severity levels, when dynamic link throttling is enabled.

A Throttling Window is defined as a period of time where the duty cycle of throttling can be specified. A Throttling Window is sub-divided into a Throttling Zone and a Non-Throttling Zone. The period of the Throttling Zone is configurable through the TNPT.TT field. Depending on the throttle severity levels, the throttling duration specified by the TNPT.TT field will be multiplied by the multipliers configurable through TNPT.TSLxM.

The period of the Throttling Window is configurable through the TNPT.TP field. The Throttling Window is always referenced from the time a new Throttle State change indication is received by the Root Port or from the time the throttling is enabled by the configuration register. The Throttling Window and Throttling Zone timers continue to behave the same as in L0 or L0s even if the link transitions to other LTSSM states, except for L1, L23_Rdy and link down. For L1 case, the timer is allowed to be stopped and hardware is allowed to re-start the Throttling Window and the corresponding Throttling Zone timers on exit from L1.

22.5.4 Port 8xh Decode

The PCIe* root ports will explicitly decode and claim I/O cycles within the 80h – 8Fh range when MPC.P8XDE is set. The claiming of these cycles are not subjected to standard PCI I/O Base/Limit and I/O Space Enable fields. This allows a POST-card to be connected to the Root Port either directly as a PCI Express device or through a PCI Express to PCI bridge as a PCI card.

Any I/O reads or writes will be forwarded to the link as it is. The device will need to be able to return the previously written value, on I/O read to these ranges. BIOS must ensure that at any one time, no more than one Root Port is enabled to claim Port 8xh cycles.

22.5.5 Separate Reference Clock with Independent SSC (SRIS)

The current PCI-SIG "PCI Express* External Cabling Specification" (www.pcisig.com) defines the reference clock as part of the signals delivered through the cable. Inclusion of the reference clock in the cable requires an expensive shielding solution to meet EMI requirements.

The need for an inexpensive PCIe* cabling solution for PCIe* SSDs requires a cabling form factor that supports non-common clock mode with spread spectrum enabled, such that the reference clock does not need to be part of the signals delivered through the cable. This clock mode requires the components on both sides of a link to tolerate a much higher ppm tolerance of ~5600 ppm compared to the PCIe* Base Specification defined as 600 ppm.

Soft straps are needed as a method to configure the port statically to operate in this mode. This mode is only enabled if the SSD connector is present on the motherboard, where the SSD connector does not include the reference clock. No change is being made to PCIe* add-in card form factors and solutions.

ASPM L0s is not supported in this form factor. The L1 exit latency advertised to software would be increased to 10 us. The root port does not support Lower SKP Ordered Set generation and reception feature defined in SRIS ECN.

22.5.6 Advanced Error Reporting

The PCI Express* Root Ports each provide basic error handling, as well as Advanced Error Reporting (AER) as described in the latest PCI Express Base Specification

22.5.7 Single- Root I/O Virtualization (SR- IOV)

Alternative Routing ID Interpretation (ARI) and Access Control Services (ACS) are supported as part of the complementary technologies to enable SR-IOV capability.

22.5.7.1 Alternative Routing- ID Interpretation (ARI)

Alternative Routing-ID Interpretation (ARI) is a mechanism that can be used to extend the number of functions supported by a multi-function ARI device connected to the Root Port, beyond the conventional eight functions.

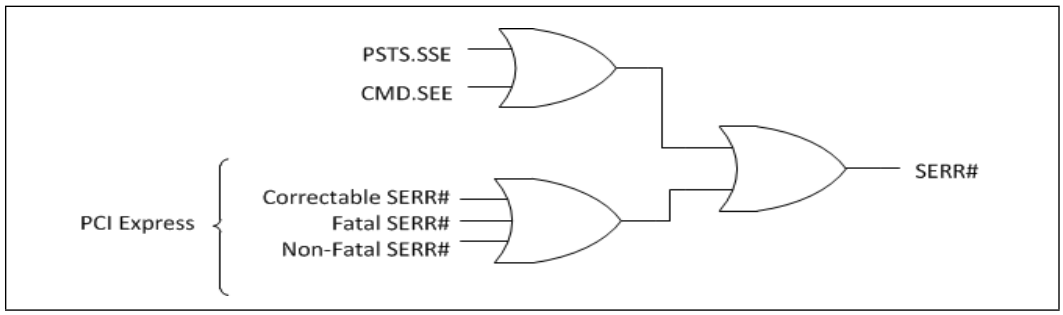
22.5.7.2 Access Control Services (ACS)

ACS is defined to control access between different Endpoints and between different Functions of a multi-function device. ACS defines a set of control points to determine whether a TLP should be routed normally, blocked, or redirected.

22.5.8 SERR# Generation

SERR# may be generated using two paths—through PCI mechanisms involving bits in the PCI header, or through PCI Express* mechanisms involving bits in the PCI Express capability structure.

Figure 22-1. Generation of SERR# to Platform



22.5.9 Hot-Plug

All PCIe* Root Ports support Express Card 1.0 based hot-plug that performs the following:

- Presence Detect and Link Active Changed Support
- Interrupt Generation Support

22.5.9.1 Presence Detection

When a module is plugged in and power is supplied, the physical layer will detect the presence of the device, and the root port sets SLSTS.PDS and SLSTS.PDC. If SLCTL.PDE and SLCTL.HPE are both set, the root port will also generate an interrupt.

When a module is removed (using the physical layer detection), the root port clears SLSTS.PDS and sets SLSTS.PDC. If SLCTL.PDE and SLCTL.HPE are both set, the root port will also generate an interrupt.

22.5.9.2 SMI/SCI Generation

Interrupts for power-management events are not supported on legacy operating systems. To support power-management on non-PCI Express aware operating systems, power management events can be routed to generate SCI. To generate SCI, MPC.HPCE must be set. When set, enabled hot-plug events will cause SMSCS.HPCS to be set.

Additionally, BIOS workarounds for hot-plug can be supported by setting MPC.HPME. When this bit is set, hot-plug events can cause SMI status bits in SMSCS to be set. Supported hot-plug events and their corresponding SMSCS bit are:

- Presence Detect Changed – SMSCS.HPPDM
- Link Active State Changed – SMSCS.HPLAS

When any of these bits are set, SMI# will be generated. These bits are set regardless of whether interrupts or SCI is enabled for hot-plug events. The SMI# may occur concurrently with an interrupt or SCI.

22.5.10 PCI Express* Lane Polarity Inversion

The PCI Express* Base Specification requires polarity inversion to be supported independently by all receivers across a Link—each differential pair within each Lane of a PCIe* Link handles its own polarity inversion. Polarity inversion is applied, as needed, during the initial training sequence of a Lane. In other words, a Lane will still function

correctly even if a positive (Tx+) signal from a transmitter is connected to the negative (Rx-) signal of the receiver. Polarity inversion eliminates the need to untangle a trace route to reverse a signal polarity difference within a differential pair and no special configuration settings are necessary in the PCH to enable it. It is important to note that polarity inversion does not imply direction inversion or direction reversal; that is, the Tx differential pair from one device must still connect to the Rx differential pair on the receiving device, per the PCIe* Base Specification. Polarity Inversion is not the same as "PCI Express* Controller Lane Reversal".

22.5.11 PCI Express* Controller Lane Reversal

For each PCIe* Controller we support end-to-end lane reversal across the four lanes mapped to a controller for the two motherboard PCIe* configurations listed below. Lane Reversal means that the most significant lane of a PCIe* Controller is swapped with the least significant lane of the PCIe* Controller while the inner lanes get swapped to preserve the data exchange sequence (order).

Note: Lane Reversal Supported Motherboard PCIe* Configurations = 1x4, 2x1+1x2, and 2x2

- The 2x1+1x2 configuration is enabled by setting the PCIe* Controller soft straps to 1x2+2x1 with Lane Reversal Enabled

Note: PCI Express* Controller Lane Reversal is not the same as PCI Express* Lane Polarity Inversion

22.5.12 Precision Time Measurement (PTM)

Hardware protocol for precise coordination of events and timing information across multiple upstream and downstream devices using Transaction Layer Protocol (TLP) Message Requests. Minimizes timing translation errors resulting in the increased coordination of events across multiple components with very fine precision.

All of the PCH PCIe* Controllers and their assigned Root Ports support PTM where each Root Port can have PTM enabled or disabled individually from one another.

Note: PCIe Root Ports transmit the lower byte [7:0] of the Propagation Delay Field first instead of the upper byte [31:24] within their PTM DelayResponseD (Response with Data) messages.

§ §

23 Power Management

23.1 Acronyms

Acronyms	Description
PMC	Power Management Controller
STD	Suspend To Disk
STR	Suspend To RAM
PMIC	Power Management Integrated Circuit
VR	Voltage Regulator

23.2 References

Specification	Location
Advanced Configuration and Power Interface (ACPI)	http://www.acpi.info/spec.htm

23.3 Overview

The Power Management Controller (PMC) is the PCH unit that handles all PCH power management related activities. This unit administers power management functions of the PCH including interfacing with other logic and controllers on the platform to perform power state transitions (such as SLP_S3# and PLTRST#); configure, manage and respond to wake events; aggregate and report latency tolerance information for devices and peripherals connected to and integrated into the PCH.

Note: DeepS3 is not supported in the PCH and in this document Deep Sx refers to DeepS4 and DeepS5 states.

23.4 Signal Description

Name	Type	Description
ACPRESENT/GPD1	I	**ACPRESENT:** This input pin indicates when the platform is plugged into AC power or not. In addition to the previous Intel® CSME to EC communication, the PCH uses this information to implement the Deep Sx policies. For example, the platform may be configured to enter Deep Sx when in S4 or S5 and only when running on battery. This is powered by Deep Sx Well.
BATLOW#/GPD0	I	**Battery Low:** An input from the battery to indicate that there is insufficient power to boot the system. Assertion will prevent wake from S4/S5 states and exit from Deep Sx state. This signal can also be enabled to cause an SMI# when asserted. This signal is multiplexed with GPD0. For any platform not using this pin functionality, this signal must be tied high to VCCDSW_3P3. ***Note:*** An external pull-up resistor to VCCDSW_3P3 is required.
CORE_VID0/ GPP_B0	O	**PCH Core VID Bit 0**: May connect to discrete VR on platform and used to control the VCCIN_AUX rail (FIVR input) voltage. **In default mode this pin is driven high ('1')**

Name	Type	Description
CORE_VID1/ GPP_B1	O	**PCH Core VID Bit 1**: May connect to discrete VR on platform and used to control the VCCIN_AUX rail (FIVR input) voltage. **In default mode this pin is driven high ('1')**
CPU_C10_GATE# / GPP_H18	O	External Power Gate control for VCCIO, VCCSTG and VCCPLL_OC during C10. When asserted, VCCIO and VCCSTG can be 0V, however the power good indicators for these rails must remain asserted.
DRAM_RESET#	OD	**System Memory DRAM Reset**: Active low reset signal to DRAM. ***Note:*** An external pull-up to the DRAM power plane is required.
DSW_PWROK	I	**DSW PWROK**: Power OK Indication for the VCCDSW_3p3 voltage rail. ***Note:*** This signal is in the RTC well. ***Note:*** This signal cannot tie together with RSMRST#.
LAN_WAKE#/GPD2	I	**LAN WAKE**: is an active low wake indicator from the GbE PHY. ***Note:*** External pull-up required.
LANPHYPC /GPD11	O	**LAN PHY Power Control:** LANPHYPC is used to indicate that power needs to be restored to the Platform LAN Connect Device, when implementing Intel Auto Detect Battery Saver feature.
PCH_PWROK	I	**PCH Power OK:** When asserted, PCH_PWROK is an indication to the PCH that all of its core power rails have been stable for at least 5 ms. PCH_PWROK can be driven asynchronously. When PCH_PWROK is negated, the PCH asserts PLTRST#. ***Note:*** PCH_PWROK must not glitch, even if RSMRST# is low.
PLTRST#/GPP_B13	O	**Platform Reset:** The PCH asserts PLTRST# to reset devices on the platform (such as SIO, LAN, processor, and so forth.). The PCH asserts PLTRST# low in Sx states and when a cold, warm, or global reset occurs. The PCH de-asserts PLTRST# upon exit from Sx states and the aforementioned resets. There is no guaranteed minimum assertion time for PLTRST#. ***Note:*** PCIe* specification requires that the power rails associated with PCIe* (typically the 3.3V, 5V, and 12V core well rails) have been valid for 100 ms prior to PLTRST# de-assertion. System designers must ensure the requirement is met on the platform.
PWRBTN#/GPD3	I	**Power Button:** The Power Button will cause SMI# or SCI to indicate a system request to go to a sleep state. If the system is already in a sleep state, this signal will cause a wake event. If PWRBTN# is pressed for more than 4 seconds (default; timing is configurable), this will cause an unconditional transition (power button override) to the S5 state. Override will occur even if the system is in the S3-S4 states. This signal has an internal Pull-up resistor and has an internal 16 ms de-bounce on the input. ***Note:*** Upon entry to S5 due to a power button override, if Deep Sx is enabled and conditions are met, the system will transition to Deep S5.
RSMRST#	I	**Primary Well Reset:** This signal is used for resetting the resume power plane logic. This signal must be asserted for at least 10ms after the suspend power wells are valid. When de-asserted, this signal is an indication that the primary power wells are stable.
SLP_A#/GPD6	O	**SLP_A#:** Signal asserted when the Intel® CSME platform goes to M-Off or M3-PG. Depending on the platform, this pin may be used to control power to various devices that are part of the Intel® CSME sub-system in the platform. If you're not using this pin for any functional purposes on your platform, or can tolerate lack of minimum assertion time, program the stretching value to the minimum.
SLP_LAN#	O	**LAN Sub-System Sleep Control:** When SLP_LAN# is de-asserted it indicates that the PHY device must be powered. When SLP_LAN# is asserted, power can be shut off to the PHY device. SLP_LAN# will always be de-asserted in S0 and anytime SLP_A# is de-asserted.

Name	Type	Description
SLP_WLAN# / GPD9	O	**WLAN Sub-System Sleep Control:** When SLP_WLAN# is asserted, power can be shut off to the external wireless LAN device. SLP_WLAN will always will be de-asserted in S0. If you're not using this pin for any functional purposes on your platform, or can tolerate lack of minimum assertion time, program the stretching value to the minimum.
SLP_S0#/GPP_B12	O	**S0 Sleep Control**: When PCH is idle and processor is in C10 state, this pin will assert to indicate VR controller can go into a light load mode. This signal can also be connected to EC for other power management related optimizations.
SLP_S3#/GPD4	O	**S3 Sleep Control:** SLP_S3# is for power plane control. This signal shuts off power to all non-critical systems when in S3 (Suspend To RAM), S4 (Suspend to Disk), or S5 (Soft Off) states.
SLP_S4#/GPD5	O	**S4 Sleep Control**: SLP_S4# is for power plane control. This signal shuts power to all non-critical systems when in the S4 (Suspend to Disk) or S5 (Soft Off) state. ***Note:*** This pin must be used to control the DRAM power in order to use the PCH DRAM power-cycling feature.
SLP_S5#/GPD10	O	**S5 Sleep Control:** SLP_S5# is for power plane control. This signal is used to shut power off to all non-critical systems when in the S5 (Soft Off) states.
SLP_SUS#	O	**Deep Sx Indication:** When asserted (driven low), this signal indicates PCH is in Deep Sx state where internal Sus power is shut off for enhanced power saving. When de-asserted (driven high), this signal indicates exit from Deep Sx state and Sus power can be applied to PCH. For non-Deep SX, this pin also needs to use to turn on the VCCPRIM_1P8 VR. This pin cannot left unconnected. ***Note:*** This pin is in the DSW power well.
SUSACK#/GPP_C7 / SML1DATA	I	**SUSACK#:** If Deep Sx is supported, the EC/motherboard controlling logic must change SUSACK# to match SUSWARN# once the EC/motherboard controlling logic has completed the preparations discussed in the description for the SUSWARN# pin. ***Note:*** SUSACK# is only required to change in response to SUSWARN# if Deep Sx is supported by the platform.
SUSCLK/GPD8	O	**Suspend Clock:** This clock is a digitally buffer version of the RTC clock.
SUSWARN#/ SUSPWRDNACK/ GPP_C6 / SML1CLK	O	**SUSWARN#:** This pin asserts low when the PCH is planning to enter the Deep Sx power state and remove Primary power (using SLP_SUS#). The EC/motherboard controlling logic must observe edges on this pin, preparing for SUS well power loss on a falling edge and preparing for Primary well related activity (host/Intel® CSME wakes and runtime events) on a rising edge. SUSACK# must be driven to match SUSWARN# once the above preparation is complete. SUSACK# should be asserted within a minimal amount of time from SUSWARN# assertion as no wake events are supported if SUSWARN# is asserted but SUSACK# is not asserted. Platforms supporting Deep Sx, but not wishing to participate in the handshake during wake and Deep Sx entry may tie SUSACK# to SUSWARN#. This pin is multiplexed with SUSPWRDNACK since it is not needed in Deep Sx supported platforms.
SUSPWRDNACK/ SUSWARN# GPP_C6 / SML1CLK	O	**SUSPWRDNACK:** Active high. Asserted by the PCH on behalf of the Intel® CSME when it does not require the PCH Primary well to be powered. Platforms are not expected to use this signal when the PCH Deep Sx feature is used.
SX_EXIT_HOLDOFF#/ GPP_H3/ CNV_BT_I2S_SDO	I	**Sx Exit Holdoff Delay**: Delay exit from Sx state after SLP_A# is de-asserted. Refer Section 23.7.7.5 for more details. Note : When eSPI is enabled, SX_EXIT_HOLDOFF# functionality is not available, and assertion of the signal will not impact Sx exit flows.
SYS_PWROK	I	**System Power OK:** This generic power good input to the PCH is driven and utilized in a platform-specific manner. While PCH_PWROK always indicates that the core wells of the PCH are stable, SYS_PWROK is used to inform the PCH that power is stable to some other system component(s) and the system is ready to start the exit from reset.

Name	Type	Description
SYS_RESET#	I	**System Reset**: This pin forces an internal reset after being de-bounced.
VRALERT#/GPP_B2	I	**VR Alert:** ICC Max. throttling indicator from the PCH voltage regulators. VRAlert# pin allows the VR to force throttling to prevent an over current shutdown. PMC_Tstate[1:0] is a new bus generated by the PMC based on the VRalert# and messages from CPU. The messages from CPU allows CPU to constrain the PCH to a particular power budget.
WAKE#	I/OD	**PCI Express* Wake Event in Sx:** Input Pin in Sx. Sideband wake signal on PCI Express* asserted by components requesting wake up. ***Note:*** This is Output pin during S0ix states hence this pin can not be used to wake up the system during S0ix states. ***Note:*** External pull-up required.
PMCALERT#/ GPP_B11	I/OD	PMC Alert Pin is added to support USB-C* "PD controller friendly" architecture.
VCCST_OVERRIDE	O	Signal that allows PCH to keep VCCST powered ON (in case VCCST is powered down) for Type-C* wake capability.
VCCST_PWRGD	O	VCCST power good to the processor as this rail is now supplied by the integrated FIVR in PCH.
INPUT3VSEL	I	Strapped high if PCH's VCCDSW_3P3 rail is 3.0V +/-5%; else PCH's VCCDSW_3P3 rail is 3.3V +/- 5%. This pin is in the VCCPRIM_3P3 well. ***Note:*** When strapped for 3.0V operation, it is expected that the rest of the platform's 3.3V rails are at 3.0V (example, the battery is a 1S configured battery) and that components can function properly at 3.0V.

23.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
ACPRESENT/GPD1	Pull-down	15 kohm - 40 kohm	1
LAN_WAKE#/GPD2	Pull-down	15 kohm – 40 kohm	1
PWRBTN#/GPD3	Pull-up	20 kohm +/- 30%	
SUSACK#/GPP_C7 / SML1DATA	Pull-up	20 kohm+/- 30%	
WAKE#	Pull-down	15 kohm – 40 kohm	1

Notes:
1. Pull-down is configurable and can be enabled in Deep Sx state; refer to DSX_CFG register for more details.

23.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[18]	Immediately after Reset[18]	S3/S4/S5	Deep Sx
ACPRESENT[6,10,15]	DSW	Undriven / Driven Low[4]	Undriven	Undriven	Undriven/ Internal Pull-down[8]
BATLOW#	DSW	Undriven	Undriven	Undriven	OFF
CORE_VID0[11,17]	Primary	Driven High	Driven High	Driven High	OFF
CORE_VID1[11,17]	Primary	Driven High	Driven High	Driven High	OFF
CPU_C10_GATE#[1,17]	Primary	Driven High	Driven High	Driven High	OFF
DRAM_RESET#[14]	DSW	Undriven	Undriven	Undriven	Undriven

Signal Name	Power Plane	During Reset[18]	Immediately after Reset[18]	S3/S4/S5	Deep Sx
DSW_PWROK	RTC	Undriven	Undriven	Undriven	Undriven
INPUT3VSEL	DSW	Undriven	Undriven	Undriven	Undriven
LANPHYPC[10,16]	DSW	Driven Low	Driven Low	Driven Low	Driven Low
LAN_WAKE#[15]	DSW	Undriven	Undriven	Undriven	Undriven/ Internal Pull-down [8]
PCH_PWROK	RTC	Undriven	Undriven	Undriven	Undriven
PLTRST#[16]	Primary	Driven Low	Driven High	Driven Low	OFF
PWRBTN#[15]	DSW	Internal Pull-up	Internal Pull-up	Internal Pull-up	Internal Pull-up
RSMRST#	RTC	Undriven	Undriven	Undriven	Undriven
SLP_S0#[1]	Primary	Driven High	Driven High	Driven High	OFF
SLP_S3#[6,16]	DSW	Driven Low	Driven High	Driven Low	Driven Low
SLP_S4#[6,16]	DSW	Driven Low	Driven High	Driven High/ Driven Low[2]	Driven High/ Driven Low[9]
SLP_S5#[6,16]	DSW	Driven Low	Driven High	Driven High/ Driven Low[3]	Driven High/ Driven Low[9]
SLP_LAN#[6,14]	DSW	Driven Low	Driven Low	Driven High/ Driven Low[7]	Driven High/ Driven Low[7]
SLP_WLAN#[6,16]	DSW	Driven Low	Driven Low	Driven High/ Driven Low[7]	Driven High/ Driven Low[7]
SLP_A#[6,16]	DSW	Driven Low	Driven High	Driven High/ Driven Low[12]	Driven High/ Driven Low[12]
SLP_SUS#[6,14]	DSW	Driven Low	Driven High	Driven High	Driven Low
SUSCLK[10,16]	DSW	Driven Low	Toggling	Toggling	Toggling[10]
SUSACK#[15]	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up	OFF
SUSWARN#/ SUSPWRDNACK[6,10, 16]	Primary	Driven Low	Driven Low	Driven Low[5]	OFF
SX_EXIT_HOLDOFF#[15]	Primary	Undriven	Undriven	Undriven	OFF
SYS_PWROK[13]	Primary	Undriven	Undriven	Undriven	OFF
SYS_RESET#[13]	Primary	Undriven	Undriven	Undriven	OFF
VR_ALERT#[15]	Primary	Undriven	Undriven	Undriven	OFF
WAKE#[13]	DSW	Undriven	Undriven	Undriven	Undriven/ Internal Pull-down

Signal Name	Power Plane	During Reset[18]	Immediately after Reset[18]	S3/S4/S5	Deep Sx

Notes:
1. Driven High during S0 and driven Low during S0 CS.
2. SLP_S4# is driven high in S3, driven low in S4/S5.
3. SLP_S5# is driven high in S3/S4, driven low in S5.
4. In non-Deep Sx mode, pin is driven low.
5. Based on wake events and Intel® CSME state. SUSPWRDNACK is always '0' while in M0 or M3, but can be driven to '0' or '1' while in Moff state. SUSPWRDNACK is the default mode of operation. If Deep Sx is supported, then subsequent boots will default to SUSWARN#.
6. The pin requires glitch-free output sequence. The pad should only be pulled low momentarily when the corresponding buffer power supply is not stable.
7. Based on wake event and Intel® CSME state.
8. Pull-down is configurable and can be enabled in Deep Sx state; refer to DSX_CFG register for more details.
9. When platform enters Deep Sx, the SLP_S4# and SLP_S5# pin will retain the value it held prior to Deep Sx entry.
10. Internal weak pull-down resistor is enabled during power sequencing, but configurable (pull-up/pull-down/none) after boot.
11. The CORE_VID pins defaults to '1' and will be driven to '1' to reflect that VCCPRIM_CORE voltage will support 1.8 V. The VID able to change to 1.8 V/ 1.65 V/ 1.1 V/ 0 V based on the CPU and the state.
12. Pin state is a function of whether the platform is configured to have Intel® CSME on or off in Sx.
13. Output High-Z, not glitch free.
14. Output High-Z, glitch free with ~1 kΩ Pull-down during respective power sequencing
15. Output High-Z, not glitch free.
16. Output High-Z, glitch free with ~20 kΩ Pull-down during respective power sequencing.
17. Output High-Z, glitch free with ~20 kΩ Pull-up during respective power sequencing.
18. Reset reference for primary well pins is RSMRST#, DSW well pins is DSW_PWROK, and RTC well pins is RTCRST#.

23.7 Functional Description

23.7.1 Features

- Support for *Advanced Configuration and Power Interface (ACPI)* providing power and thermal management
  - — ACPI 24-Bit Timer SCI and SMI# Generation
- PCI PME# signal for Wake Up from Low-Power states
- System Sleep State Control
  - — ACPI S3 state – Suspend to RAM (STR)
  - — ACPI S4 state – Suspend-to-Disk (STD)
  - — ACPI G2/S5 state – Soft Off (SOFF)
  - — Power Failure Detection and Recovery
  - — Deep Sx
- Intel Management Engine Power Management Support
  - — Wake events from the Intel Management Engine (enabled from all S-States including Catastrophic S5 conditions)
- SLP_S0# signal for external platform VR power gating or EC power management handling during lower power condition

23.7.2 PCH S0 Low Power

The PCH has many independent functions and I/O interfaces making power management a highly distributive task. The first level of power management is to control the independent resources and the best place to do that is in the controllers. The second level of power management is to control the shared resources, which requires communication amongst the users of the shared resources.

The PCH power states are a combination of first level and second level power management functions. The "deeper" the power state, meaning the lower power required, generally means that more resources are disabled.

23.7.2.1 PCH S0 Low Power State Definition

A high level description of the global PCH low power states are described in below table. This table does not discuss the conditions to enter into these states, only the summary of the PCH power actions that are taken. These states are also not rigid definitions of actual HW states meaning that there are not specific flows to enter into LPx states. Most of the power management on the PCH is done autonomously by the I/O interface's controller and is not globally controlled.

Table 23-1. PCH Low Power State

Power State	Description	CPU Package State	Power Action
LP1	Fully running S0 with aggressive opportunistic power management actions	C0	• OPI L1 and PLL shutdown • Individual PLL shutdown[1] • Internal power gating of PCH controllers[2] • Internal HSIO per lane power gating
LP2	Pervasively Idle S0 and Root PLLs are off	C6 or deeper	All actions from LP1 + • Gen 2 PLL/BCLK PLL shutdown
LP3	Idle Floor	C10	All actions from LP2 + • XTAL shutdown • SLP_S0# • VCCPRIM_CORE Low Voltage Mode

Notes:
1. Individual PLL shutdown – Each I/O interface when becoming sufficiently idle (typically requiring a minimum link power state) can have its respective I/O PLL be shutdown dynamically. This includes PCIe Gen3, SATA, USB 2.0 and MIPI.
2. Internal Power Gating of PCH controllers – Each host controller (that is, xHCI, AHCI), PCIe* root port or embedded subsystem (ISH, Intel® CSME, Audio) when becoming sufficiently idle can autonomously power gate its core digital logic and local memory arrays. xHCI power gating is on a per port basis.

23.7.3 PCH and System Power States

Table 23-2 shows the power states defined for PCH-based platforms. The state names generally match the corresponding ACPI states.

Table 23-2. General Power States for Systems Using the PCH (Sheet 1 of 2)

State / Substates	Legacy Name/Description
G0/S0/C0	**Full On**: Processor operating. Individual devices may be shut down or be placed into lower power states to save power.

Table 23-2. General Power States for Systems Using the PCH (Sheet 2 of 2)

State / Substates	Legacy Name/Description
G0/S0/Cx	**Cx State**: Cx states are processor power states within the S0 system state that provide for various levels of power savings. The processor manages c-state itself. The actual c-state is not passed to the PCH. Only c-state related messages are sent to the PCH and PCH will base its behavior on the actual data passed.
G1/S3	**Suspend-To-RAM (STR):** The system context is maintained in system DRAM, but power is shut off to non-critical circuits. Memory is retained and refreshes continue. All external clocks stop except RTC.
G1/S4	**Suspend-To-Disk (STD):** The context of the system is maintained on the disk. All power is then shut off to the system except for the logic required to resume.
G2/S5	**Soft Off (SOFF):** System context is not maintained. All power is shut off except for the logic required to restart. A full boot is required when waking.
S0ix	S0 idle states are often referred as S0ix states. CPU PKG C-states and platform latency tolerance will decide when to take the aggressive power management actions.
Deep Sx	**Deep Sx:** An optional low power state where system context may or may not be maintained depending upon entry condition. All power is shut off except for minimal logic that allows exiting Deep Sx. If Deep Sx state was entered from S4 state, then the resume path will place system back into S4. If Deep Sx state was entered from S5 state, then the resume path will place system back into S5.
G3	**Mechanical OFF (M-Off):** System context not maintained. All power is shut off except for the RTC. No "Wake" events are possible. This state occurs if the user removes the main system batteries in a mobile system, 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 will depend on the state just prior to the entry to G3 and the AFTERG3_EN bit in the General Power Management Configuration (GEN_PMCON). Refer to Table 23-8 for more details.

Table 23-3 shows the transitions rules among the various states.

Note: Transitions among the various states may appear to temporarily transition through intermediate states. For example, in going from S0 to S4, it may appear to pass through the G1/S3 state. These intermediate transitions and states are not listed in the Table 23-3.

Table 23-3. State Transition Rules for the PCH (Sheet 1 of 2)

Present State	Transition Trigger	Next State
G0/S0/C0	• OPI Msg • SLP_EN bit set • Power Button Override[3,5] • Mechanical Off/Power Failure	• G0/S0/Cx • G1/Sx or G2/S5 state • G2/S5 • G3
G0/S0/Cx	• OPI Msg • Power Button Override[3,5] • Mechanical Off/Power Failure	• G0/S0/C0 • S5 • G3
G1/S3	• Any Enabled Wake Event • Power Button Override[3,5] • Mechanical Off/Power Failure	• G0/S0/C0[2] • G2/S5 • G3
G1/S4	• Any Enabled Wake Event • Power Button Override[3,5] • Conditions met as described in Section 23.7.6.6.1 and Section 23.7.6.6.2 • Mechanical Off/Power Failure	• G0/S0/C0[2] • G2/S5 • Deep S4 • G3

Table 23-3. State Transition Rules for the PCH (Sheet 2 of 2)

Present State	Transition Trigger	Next State
G2/S5	• Any Enabled Wake Event • Conditions met as described in Section 23.7.6.6.1 and Section 23.7.6.6.2 • Mechanical Off/Power Failure	• G0/S0/C0[2] • Deep S5 • G3
G2/Deep Sx	• Any Enabled Wake Event • ACPRESENT Assertion • Mechanical Off/Power Failure • Power Button Override	• G0/S0/C0[2] • G1/S4 or G2/S5 (Refer Section 23.7.6.6.2) • G3 • G2/S5
G3	• Power Returns	• S0/C0 (reboot) or G2/S5[4] (stay off until power button pressed or other wake event)[1,2]

Notes:
1. Some wake events can be preserved through power failure.
2. Transitions from the S3–S5 or G3 states to the S0 state are deferred until BATLOW# is inactive in mobile configurations.
3. Includes all other applicable types of events that force the host into and stay in G2/S5.
4. If the system was in G1/S4 before G3 entry, then the system will go to S0/C0 or G1/S4.
5. Upon entry to S5 due to a power button override, if Deep S5 is enabled and conditions are met per Section 23.7.6.6, the system will transition to Deep S5.

23.7.3.1 System Power Planes

The system has several independent power planes, as described in Table 23-4.

Note: When a particular power plane is shut off, it should go to a 0 V level.

Table 23-4. System Power Plane (Sheet 1 of 2)

Plane	Controlled By	Description
CPU	SLP_S3# signal	The SLP_S3# signal can be used to cut the power to the processor completely.
Main (Applicable to Platform, PCH does not have a Main well)	SLP_S3# signal	When SLP_S3# goes active, power can be shut off to any circuit not required to wake the system from the S3 state. Since the S3 state requires that the memory context be preserved, power must be retained to the main memory. The processor, PCI Express* will typically be power-gated when the Main power plane is shut, although there may be small subsections powered. ***Note:*** The PCH power is not controlled by the SLP_S3# signal, but instead by the SLP_SUS# signal.
Memory	SLP_S4# signal SLP_S5# signal	When SLP_S4# goes active, power can be shut off to any circuit not required to wake the system from the S4. Since the memory context does not need to be preserved in the S4 state, the power to the memory can also be shut down. When SLP_S5# goes active, power can be shut off to any circuit not required to wake the system from the S5 state. Since the memory context does not need to be preserved in the S5 state, the power to the memory can also be shut.
Intel® CSME	SLP_A#	SLP_A# signal is asserted when the Intel® CSME platform goes to M-Off or M3-PG. Depending on the platform, this pin may be used to control power to various devices that are part of the Intel® CSME sub-system in the platform.
LAN	SLP_LAN#	This signal is asserted in Sx/M-Off or Sx/M3-PG when both host and Intel® CSME WoL are not supported. This signal can be use to control power to the Intel GbE PHY.
Primary/ Suspend Well	SLP_SUS#	This signal is asserted when the Primary/Suspend rails can be externally shut off for enhanced power saving.

Table 23-4. System Power Plane (Sheet 2 of 2)

Plane	Controlled By	Description
VCCIO and VCCSTG	CPU_C10_GATE#	This signal is asserted when the processor enters C10 and can handle VCCIO, VCCSTG and VCCPLL_OC being lowered to 0V.
DEVICE[n]	Implementation Specific	Individual subsystems may have their own power plane. For example, GPIO signals may be used to control the power to disk drives, audio amplifiers, or the display screen.

23.7.4 SMI#/SCI Generation

Upon any enabled SMI event taking place while the End of SMI (EOS) bit is set, the PCH will clear the EOS bit and assert SMI to the processor, which will cause it to enter SMM space. SMI assertion is performed using a Virtual Legacy Wire (VLW) message. Prior system generations (those based upon legacy processors) used an actual SMI# pin.

Once the SMI VLW has been delivered, the PCH takes no action on behalf of active SMI events until Host software sets the End of SMI (EOS) bit. At that point, if any SMI events are still active, the PCH will send another SMI VLW message.

The SCI is a level-mode interrupt that is typically handled by an ACPI-aware operating system. In non-APIC systems (which is the default), the SCI IRQ is routed to one of the 8259 interrupts (IRQ 9, 10, or 11). The 8259 interrupt controller must be programmed to level mode for that interrupt.

In systems using the APIC, the SCI can be routed to interrupts 9, 10, 11, 20, 21, 22, or 23. The interrupt polarity changes depending on whether it is on an interrupt shareable with a PIRQ or not. The interrupt remains asserted until all SCI sources are removed.

Below table shows which events can cause an SMI and SCI.

Note: Some events can be programmed to cause either an SMI or SCI. The usage of the event for SCI (instead of SMI) is typically associated with an ACPI-based system. Each SMI or SCI source has a corresponding enable and status bit.

Table 23-5. Causes of SMI and SCI (Sheet 1 of 3)

Cause	SCI	SMI	Additional Enables (Note 1)	Where Reported
PME#	Yes	Yes	PME_EN=1	PME_STS
PME_B0 (Internal, Bus 0, PME-Capable Agents)	Yes	Yes	PME_B0_EN=1	PME_B0_STS
PCI Express* PME Messages	Yes	Yes	PCI_EXP_EN=1 (Not enabled for SMI)	PCI_EXP_STS
PCI Express* Hot-Plug Message	Yes	Yes	HOT_PLUG_EN=1 (Not enabled for SMI)	HOT_PLUG_STS
Power Button Press	Yes	Yes	PWRBTN_EN=1	PWRBTN_STS
Power Button Override (Note 6)	Yes	No	None	PWRBTNOR_STS
RTC Alarm	Yes	Yes	RTC_EN=1	RTC_STS
ACPI Timer overflow (2.34 seconds)	Yes	Yes	TMROF_EN=1	TMROF_STS
GPIO	Yes	Yes	Refer Note 8	
LAN_WAKE#	Yes	Yes	SCI_EN=0, LAN_WAKE_EN=1	LAN_WAKE_STS
TCO SCI message from processor	Yes	No	None	CPUSCI_STS

Table 23-5. Causes of SMI and SCI (Sheet 2 of 3)

Cause	SCI	SMI	Additional Enables (Note 1)	Where Reported
TCO SCI Logic	Yes	No	TCOSCI_EN=1	TCOSCI_STS
TCO SMI Logic	No	Yes	TCO_EN=1	TCO_STS
TCO SMI – Year 2000 Rollover	No	Yes	None	NEWCENTURY_STS
TCO SMI – TCO TIMEROUT	No	Yes	None	TIMEOUT
TCO SMI – OS writes to TCO_DAT_IN register	No	Yes	None	OS_TCO_SMI
TCO SMI – NMI occurred (and NMIs mapped to SMI)	No	Yes	NMI2SMI_EN=1	TCO_STS, NMI2SMI_STS
TCO SMI – Changes of the WPD (Write Protect Disable) bit from 0 to 1	No	Yes	LE (Lock Enable)=1	BIOSWR_STS
TCO SMI – Write attempted to BIOS	No	Yes	WPD=0	BIOSWR_STS
BIOS_RLS written to 1 (Note 7)	Yes	No	GBL_EN=1	GBL_STS
GBL_RLS written to	No	Yes	BIOS_EN=1	BIOS_STS
Write to B2h register	No	Yes	APMC_EN = 1	APM_STS
Periodic timer expires	No	Yes	PERIODIC_EN=1	PERIODIC_STS
64 ms timer expires	No	Yes	SWSMI_TMR_EN=1	SWSMI_TMR_STS
Enhanced USB Legacy Support Event	No	Yes	LEGACY_USB2_EN = 1	LEGACY_USB2_STS
Serial IRQ SMI reported	No	Yes	None	SERIRQ_SMI_STS
Device monitors match address in its range	No	Yes	Refer DEVTRAP_STS register description	DEVTRAP_STS
SMBus Host Controller	No	Yes	SMB_SMI_EN, Host Controller Enabled	SMBus host status reg.
SMBus Slave SMI message	No	Yes	None	SMBUS_SMI_STS
SMBus SMBALERT# signal active	No	Yes	None	SMBUS_SMI_STS
SMBus Host Notify message received	No	Yes	HOST_NOTIFY_INTREN	SMBUS_SMI_STS, HOST_NOTIFY_STS
(Mobile Only) BATLOW# assertion	Yes	Yes	BATLOW_EN=1	BATLOW_STS
Access microcontroller 62h/66h	No	Yes	MCSMI_EN	MCSMI_STS
SLP_EN bit written to 1	No	Yes	SMI_ON_SLP_EN=1	SMI_ON_SLP_EN_STS
SPI Command Completed	No	Yes	None	SPI_SMI_STS
eSPI SCI/SMI Request	Yes	Yes	eSPI_SCI_EN	eSPI_SCI_STS eSPI_SMI_STS
Software Generated GPE	Yes	Yes	SWGPE_EN=1	SWGPE_STS
Intel® CSME	Yes	Yes	ME_SCI_EN=1 ME_SCI_EN=0; ME_SMI_EN=1;	ME_SCI_STS ME_SMI_STS
GPIO Lockdown Enable bit changes from '1' to '0'	No	Yes	GPIO_UNLOCK_SMI_EN=1	GPIO_UNLOCK_SMI_STS
USB 3.2 (xHCI) SMI Event	No	Yes	xHCI_SMI_EN=1	xHCI_SMI_STS
Wake Alarm Device Timer	Yes	Yes	WADT_EN	WADT_STS
ISH	Yes	No	ISH_EN	ISH_STS
RTC update-in-progress	No	Yes	Refer Vol2	RTC_UIP_SMI_STS
SIO SMI events	No	Yes	SIP_SMI_EN	SIO_SMI_STS

Table 23-5. Causes of SMI and SCI (Sheet 3 of 3)

Cause	SCI	SMI	Additional Enables (Note 1)	Where Reported
SCC	No	Yes	SCC_SMI_EN	SCC_SMI_STS

Notes:
1. SCI_EN must be 1 to enable SCI, except for BIOS_RLS. SCI_EN must be 0 to enable SMI.
2. SCI can be routed to cause interrupt 9:11 or 20:23 (20:23 only available in APIC mode).
3. GBL_SMI_EN must be 1 to enable SMI.
4. EOS must be written to 1 to re-enable SMI for the next 1.
5. The PCH must have SMI fully enabled when the PCH is also enabled to trap cycles. If SMI is not enabled in conjunction with the trap enabling, then hardware behavior is undefined.
6. When a power button override first occurs, the system will transition immediately to S5. The SCI will only occur after the next wake to S0 if the residual status bit (PRBTNOR_STS) is not cleared prior to setting SCI_EN.
7. GBL_STS being set will cause an SCI, even if the SCI_EN bit is not set. Software must take great care not to set the BIOS_RLS bit (which causes GBL_STS to be set) if the SCI handler is not in place.
8. Refer to GPIO chapter for specific GPIOs enabled for SCIs and/or SMIs

23.7.4.1 PCI Express* SCI

PCI Express* ports and the processor have the ability to cause PME using messages. When a PME message is received, the PCH will set the PCI_EXP_STS bit. If the PCI_EXP_EN bit is also set, the PCH can cause an SCI using the GPE0_STS (replaced GPE1_STS) register.

23.7.4.2 PCI Express* Hot-Plug

PCI Express* has a hot-plug mechanism and is capable of generating a SCI using the GPE0 (replaced GPE1) register. It is also capable of generating an SMI. However, it is not capable of generating a wake event.

23.7.5 C-States

PCH-based systems implement C-states by having the processor control the states. The chipset exchanges messages with the processor as part of the C-state flow, but the chipset does not directly control any of the processor impacts of C-states, such as voltage levels or processor clocking.

23.7.6 Sleep States

23.7.6.1 Sleep State Overview

The PCH directly supports different sleep states (S3–S5), which are entered by methods such as setting the SLP_EN bit or due to a Power Button press. The entry to the Sleep states is based on several assumptions:

- The G3 state cannot be entered using any software mechanism. The G3 state indicates a complete loss of power.

23.7.6.2 Initiating Sleep State

Sleep states (S3–S5) are initiated by:

- Masking interrupts, turning off all bus master enable bits, setting the desired type in the SLP_TYP field, and then setting the SLP_EN bit. The hardware then attempts to gracefully put the system into the corresponding Sleep state.
- Pressing the PWRBTN# Signal for more than 4 seconds to cause a Power Button Override event. In this case the transition to the S5 state is less graceful, since

there are no dependencies on OPI messages from the processor or on clocks other than the RTC clock.

- Assertion of the THERMTRIP# signal will cause a transition to the S5 state. This can occur when system is in S0 state.
- Shutdown by integrated manageability functions (ASF/Intel AMT)
- Internal watchdog timer Timeout events

Table 23-6. Sleep Types

Sleep Type	Comment
S3	The PCH asserts SLP_S3#. The SLP_S3# signal controls the power to non-critical circuits. Power is only retained to devices needed to wake from this sleeping state, as well as to the memory.
S4	The PCH asserts SLP_S3# and SLP_S4#. The SLP_S4# signal shuts off the power to the memory subsystem. Only devices needed to wake from this state should be powered.
S5	The PCH asserts SLP_S3#, SLP_S4# and SLP_S5#.

23.7.6.3 Exiting Sleep States

Sleep states (S3–S5) are exited based on wake events. The wake events forces the system to a full on state (S0), although some non-critical subsystems might still be shut off and have to be brought back manually. For example, the hard disk may be shut off during a sleep state and have to be enabled using a GPIO pin before it can be used.

Upon exit from the PCH-controlled Sleep states, the WAK_STS bit is set. The possible causes of wake events (and their restrictions) are shown in Table 23-7.

Note: (Mobile Only) If the BATLOW# signal is asserted, the PCH does not attempt to wake from an S3–S5 state, nor will it exit from Deep Sx state, even if the power button is pressed. This prevents the system from waking when the battery power is insufficient to wake the system. Wake events that occur while BATLOW# is asserted are latched by the PCH, and the system wakes after BATLOW# is de-asserted.

Table 23-7. Causes of Wake Events (Sheet 1 of 2)

Cause	How Enabled	Wake from Sx	Wake from Deep Sx	Wake from Sx After Power Loss (Note 2)	Wake from "Reset" Types (Note 3)
RTC Alarm	Set RTC_EN bit in PM1_EN_STS register.	Yes	Yes	Yes	No
Power Button	Always enabled as Wake event.	Yes	Yes	Yes	Yes
Any GPIOs except DSW GPIOs can be enabled for wake	Refer Note 5	Yes	No	No	No
LAN_WAKE#	Enabled natively (unless pin is configured to be in GPIO mode)	Yes	Yes	Yes	Yes
Intel® High Definition Audio	Event sets PME_B0_STS bit; PM_B0_EN must be enabled. Can not wake from S5 state if it was entered due to power failure or power button override.	Yes	No	Yes	No
Primary PME#	PME_B0_EN bit in GPE0_EN[127:96] register.	Yes	No	Yes	No
Secondary PME#	Set PME_EN bit in GPE0_EN[127:96] register.	Yes	No	Yes	No

Table 23-7. Causes of Wake Events (Sheet 2 of 2)

Cause	How Enabled	Wake from Sx	Wake from Deep Sx	Wake from Sx After Power Loss (Note 2)	Wake from "Reset" Types (Note 3)
PCI Express* WAKE# pin	PCIEXP_WAKE_DIS bit.	Yes	Yes	Yes	No
SMBALERT#	(Note 4)	Yes	No	Yes	Yes
SMBus Slave Wake Message (01h)	Wake/SMI# command always enabled as a Wake event. ***Note:*** SMBus Slave Message can wake the system from S3–S5, as well as from S5 due to Power Button Override.	Yes	No	Yes	Yes
SMBus Host Notify message received	HOST_NOTIFY_WKEN bit SMBus Slave Command register. Reported in the SMB_WAK_STS bit in the GPE0_STS register.	Yes	No	Yes	Yes
Intel® CSME Non-Maskable Wake	Always enabled as a wake event.	Yes	No	Yes	Yes
Integrated WoL Enable Override	WoL Enable Override bit (in Configuration Space).	Yes	Yes	Yes	Yes
Wake Alarm Device	WADT_EN in GPE0_EN[127:96]	Yes	Yes	No	No
AC_PRESENT	AC_PRESENT_WAKE_EN (Note 6)	No	Yes	No	No
USB connection in/after deep-Sx	GPE0_EN.USB_CON_DSX_EN+	(Note 7)	Yes	No	No

Notes:
1. If BATLOW# signal is low, PCH will not attempt to wake from S3-S5 (nor will it exit Deep Sx), even if valid wake event occurs. This prevents the system from waking when battery power is insufficient to wake the system. However, once BATLOW# goes back high, the system will boot.
2. This column represents what the PCH would honor as wake events but there may be enabling dependencies on the device side which are not enabled after a power loss.
3. Reset Types include: Power Button override, Intel® CSME-initiated power button override, Intel® CSME-initiated host partition reset with power down, Intel® CSME Watchdog Timer, SMBus unconditional power down, processor thermal trip, PCH catastrophic temperature event.
4. SMBALERT# signal is multiplexed with a GPIO pin that defaults to GPIO mode. Hence, SMBALERT# related wakes are possible only when this GPIO is configured in native mode, which means that BIOS must program this GPIO to operate in native mode before this wake is possible. Because GPIO configuration is in the resume well, wakes remain possible until one of the following occurs: BIOS changes the pin to GPIO mode, a G3 occurs or Deep Sx entry occurs.
5. There are only 72 bits in the GPE registers to be assigned to GPIOs, though any of the GPIOs can trigger a wake, only those status of GPIO mapped to 1-tier scheme are directly accessible through the GPE status registers. For those GPIO mapped under 2-tier scheme, their status would be reflected under single master status, "GPIO_TIER2_SCI_STS" or GPE0_STS and further comparison needed to know which 2-tier GPI(s) has triggered the GPIO Tier 2 SCI.
6. A change in AC_PRESENT causes an exit from Deep Sx to Sx, but the system will not wake all the way to S0.
7. Connection of a USB device can cause a wake from normal Sx as well. But that class of wakes is routed through PME_B0, not through this wake enable. The USB_CON_DSX_EN applies only to connection wakes while in Deep Sx or while in Sx after Deep Sx. Note: Sx after Deep Sx reached due to an ME wake from Deep Sx or due to AC_PRESENT going high while in Deep Sx if Deep Sx is only enabled while on DC power. The following additional conditions are required for this wake to occur:
   - The bit(s) in PM_CFG2.USB_DSX_PER_PORT_EN associated with the port(s) which experienced the connection must be set to '1'.
   - DSX_CFG.USB_CON_DSX_MODE must be set to '1', routing USB connection to generate a wake rather than be reflected out to a pin

23.7.6.4 PCI Express* WAKE# Signal and PME Event Message

PCI Express* ports can wake the platform from any sleep state (S3, S4, or S5 or Deep Sx) using the WAKE# pin. WAKE# is treated as a wake event, but does not cause any bits to go active in the GPE_STS register.

Note: PCI Express* WAKE# pin is an Output in S0ix states hence this pin cannot be used to wake up the system during S0ix states.

PCI Express* ports and the processor have the ability to cause PME using messages.These are logically OR'd to set the single PCI_EXP_STS bit. When a PME message is received, the PCH will set the PCI_EXP_STS bit. If the PCI_EXP_EN bit is also set, the PCH can cause an SCI via GPE0_STS register.

23.7.6.5 Sx-G3-Sx, Handling Power Failures

Depending on when the power failure occurs and how the system is designed, different transitions could occur due to a power failure.

The AFTERG3_EN bit provides the ability to program whether or not the system should boot once power returns after a power loss event. If the policy is to not boot, the system remains in an S5 state (unless previously in S4). There are only three possible events that will wake the system after a power failure.

1. **PWRBTN#:** PWRBTN# is always enabled as a wake event. When PCH_DPWROK is low (G3 state), the PWRBTN_STS bit is reset. When the PCH exits G3 after power returns (PCH_DPWROK goes high), the PWRBTN# signal will transition high due internal Pull-up, unless there is an on-board Pull-up/Pull-down) and the PWRBTN_STS bit is 0.
2. **RTC Alarm:** The RTC_EN bit is in the RTC well and is preserved after a power loss. Like PWRBTN_STS the RTC_STS bit is cleared when PCH_DPWROK goes low.
3. Any enabled wake event that was preserved through the power failure.

DSW_PWROK going low would place the PCH into a G3 state.

Although PME_EN is in the RTC well, this signal cannot wake the system after a power loss. PME_EN is cleared by RTCRST#, and PME_STS is cleared by RSMRST#.

Table 23-8. Transitions Due to Power Failure

State at Power Failure	AFTERG3_EN Bit	Transition when Power Returns and BATLOW# is inactive
S0, S3	1 0	S5 S0
S4	1 0	S4 S0
S5	1 0	S5 S0
Deep S4	1 0	Deep S4[1] S0
Deep S5	1 0	Deep S5[1] S0

Note:
1. Entry state to Deep Sx is preserved through G3 allowing resume from Deep Sx to take appropriate path (that is, return to S4 or S5).
2. G3 related Power Failure is defined as DSW_PWROK transition low.

23.7.6.6 Deep Sx

To minimize power consumption while in (removing S3/) S4/S5, the PCH supports a lower power, lower featured version of these power states known as Deep Sx. In the Deep Sx state, the Suspend wells are powered off, while the Deep Sx Well (DSW) remains powered. A limited set of wake events are supported by the logic located in the DSW.

The Deep Sx capability and the SUSPWRDNACK pin functionality are mutually exclusive.

23.7.6.6.1 Entry Into Deep Sx

A combination of conditions is required for entry into Deep Sx. PMC firmware is responsible for enforcing these requirements. The requirements, all of which must be met to enter Deep Sx, are detailed below:

- RTCPMCFG.INT_SUS_PD_EN = 1
  - — Intel® CSME must program this bit prior to initiating CMOFF or CM3-PG entry
- Intel® CSME in CMOFF or CM3-PG
  - — Deep Sx conditions are checked during CMOFF and CM3-PG entry. If Deep Sx entry would have been allowed if the AC_PRESENT# signal had been high, PMC FW will enable AC_PRESENT# as an interrupt source, initiating Deep Sx entry if the power source changes to match the required state
- Host in S3, S4 or S5 and combination of S-state and power source matches the host policy bits
  - — ((S3AC_GATE_SUS AND S3) OR (S4AC_GATE_SUS AND S4) OR (S5AC_GATE_SUS AND S5))1

OR

  - — ((AC_PRESENT = 0) AND ((S3DC_GATE_SUS AND S3) OR (S4DC_GATE_SUS AND S4) OR (S5DC_GATE_SUS AND S5)))
- Either Deep Sx entry is not determined by BATLOW# state or BATLOW# is asserted
  - — REQ_BATLOW_DSX == '0' OR BATLOW# == '0'
- Either Deep Sx entry is not determined by connectivity wake enable or connectivity wake is enabled
  - — REQ_CNV_NOWAKE_DSX == '0' OR SLP_WLAN_VAL == '0'

Table 23-9. Supported Deep Sx Policy Configurations

Configuration	S4DC_GATE_SUS	S4AC_GATE_SUS	S5DC_GATE_SUS	S5AC_GATE_SUS
1. Enabled in S5 Battery Only (ACPRESENT = 0)	0	0	1	0
2. Enabled in S5 (ACPRESENT not considered)	0	0	1	1
3. Enabled in S4 and S5 when on Battery only (ACPRESENT = 0)	1	0	1	0
4. Enabled in S4 and S5 (ACPRESENT not considered)	1	1	1	1
5. Enabled in S3,S4 and S5 when on Battery only (ACPRESENT = 0)	1	0	1	0
6. Enabled in S3, S4 and S5 (ACPRESENT not considered)	1	1	1	1
7. Deep S4/ S5 disabled	0	0	0	0
Note: All other configurations are RESERVED.				

The PCH also performs a SUSWARN#/SUSACK# handshake to ensure the platform is ready to enter Deep Sx. The PCH asserts SUSWARN# as notification that it is about to enter Deep Sx. Before the PCH proceeds and asserts SLP_SUS#, the PCH waits for SUSACK# to assert.

23.7.6.6.2 Exit from Deep Sx

While in Deep Sx, the PCH monitors and responds to a limited set of wake events (RTC Alarm, Power Button and WAKE#). Upon sensing an enabled Deep Sx wake event, the PCH brings up the Suspend well by de-asserting SLP_SUS#.

Table 23-10. Deep Sx Wake Events

Event	Enable
RTC Alarm	RTC_EN bit in PM1_EN_STS Register
Power Button	Always enabled
PCIe* WAKE# pin	PCIEXP_WAKE_DIS
Wake Alarm Device	WADT_EN in GPE0_EN
LAN_WAKE#	Enabled natively (unless the pin is configured to be in the GPIO mode)

ACPRESENT has some behaviors that are different from the other Deep Sx wake events. If the Intel® CSME has enabled ACPRESENT as a wake event then it behaves just like any other Intel® CSME Deep Sx wake event. However, even if ACPRESENT wakes are not enabled, if the Host policies indicate that Deep Sx is only supported when on battery, then ACPRESENT going high will cause the PCH to exit Deep Sx. In this case, the Suspend wells gets powered up and the platform remains in Sx/M-Off or Sx/M3-PG (added Sx/M-Off or Sx/M3-PD, and removed S3/M-Off, S4/M-Off or S5/M-Off). If ACPRESENT subsequently drops (before any Host or Intel® CSME wake events are detected), the PCH will re-enter Deep Sx.

23.7.7 Event Input Signals and Their Usage

The PCH has various input signals that trigger specific events. This section describes those signals and how they should be used.

23.7.7.1 PWRBTN# (Power Button)

The PCH PWRBTN# signal operates as a "Fixed Power Button" as described in the *Advanced Configuration and Power Interface Specification.* PWRBTN# signal has a 16 ms de-bounce on the input. The state transition descriptions are included in Table 23-11.

After any PWRBTN# assertion (falling edge), the 16ms de-bounce applies before the state transition starts if PB_DB_MODE='0'. If PB_DB_MODE='1', the state transition starts right after any PWRBTN# assertion (before passing through the de-bounce logic) and subsequent falling PWRBTN# edges are ignored until after 16ms.

During the time that any SLP_* signal is stretched for an enabled minimum assertion width, the host wake-up is held off. As a result, it is possible that the user will press and continue to hold the Power Button waiting for the system to wake. Unfortunately, a 4 second press of the Power Button is defined as an unconditional power down, resulting in the opposite behavior that the user was intending. Therefore, the Power Button Override Timer will be extended to 9-10 seconds while the SLP_* stretching timers are in progress. Once the stretching timers have expired, the Power Button will

awake the system. If the user continues to press Power Button for the remainder of the 9-10 seconds it will result in the override condition to S5. Extension of the Power Button Override timer is only enforced following graceful sleep entry and during host partition resets with power cycle or power down. The timer is not extended immediately following power restoration after a global reset, G3 or Deep Sx.

The PCH also supports modifying the length of time the Power Button must remain asserted before the unconditional power down occurs (4-14 seconds). The length of the Power Button override duration has no impact on the "extension" of the power button override timer while SLP_* stretching is in progress. The extended power button override period while stretching is in progress remains 9-10 seconds in all cases.

Table 23-11. Transitions Due to Power Button

Present State	Event	Transition/Action	Comment
S0/Cx	PWRBTN# goes low	SMI or SCI generated (depending on SCI_EN, PWRBTN_EN and GLB_SMI_EN)	Software typically initiates a Sleep state ***Note:*** Processing of transitions starts within 100 us of the PWRBTN# input pin to PCH going low.[1]
S3 – S5	PWRBTN# goes low	Wake Event. Transitions to S0 state	Standard wakeup ***Note:*** Could be impacted by SLP_* min assertion. The minimum time the PWRBTN# pin should be asserted is 150 us. The PCH will start processing this change once the minimum time requirement is satisfied.[1]
Deep Sx	PWRBTN# goes low	Wake Event. Transitions to S0 state	Standard wakeup ***Note:*** Could be impacted by SLP_* min assertion. The minimum time the PWRBTN# pin should be asserted is 150 us. The PCH will start processing this change once the minimum time requirement is satisfied but subsequently the PWRBTN# pin needs to de-assert for at least 500 us after RSMRST# de-assertion otherwise the system waits indefinitely in S5 state.[1]
G3	PWRBTN# pressed	None	No effect since no power Not latched nor detected ***Note:*** During G3 exit, PWRBTN# pin must be kept de-asserted for a minimum time of 500 us after the RSMRST# has de-asserted.[2] ***Note:*** Beyond this point, the minimum time the PWRBTN# pin has to be asserted to be registered by PCH as a valid wake event is 150 us.[1]
S0 – S4	PWRBTN# held low for at least 4 [3] consecutive seconds	Unconditional transition to S5 state and if Deep Sx is enabled and conditions are met per Section 23.7.6.6, the system will then transition to Deep Sx.	No dependence on processor or any other subsystem ***Note:*** Due to internal PCH latency, it could take up to an additional ~1.3s after PWRBTN# has been held low for 4s before the system would begin transitioning to S5.

Notes:
1. If PM_CFG.PB_DB_MODE='0', the debounce logic adds 16 ms to the start/minimum time for processing of power button assertions.
2. This minimum time is independent of the PM_CFG.PB_DB_MODE value.
3. The amount of time PWRBTN# must be asserted is configurable via PM_CFG2.PBOP. 4 seconds is the default.

Power Button Override Function

If PWRBTN# is observed active for at least four consecutive seconds (always sampled after the output from debounce logic), the PCH should unconditionally transition to the G2/S5 state or Deep Sx, regardless of present state (S0 – S4), even if the PCH_PWROK is not active. In this case, the transition to the G2/S5 state or Deep Sx does not depend on any particular response from the processor, nor any similar dependency from any other subsystem.

The minimum period is configurable by BIOS and defaults to the legacy value of 4 seconds.

The PWRBTN# status is readable to check if the button is currently being pressed or has been released. If PM_CFG.PB_DB_MODE='0', the status is taken after the de-bounce. If PM_CFG.PB_DB_MODE='1', the status is taken before the de-bounce. In either case, the status is readable using the PWRBTN_LVL bit.

Note: The 4-second PWRBTN# assertion should only be used if a system lock-up has occurred.

Sleep Button

The Advanced Configuration and Power Interface Specification defines an optional Sleep button. It differs from the power button in that it only is a request to go from S0 to S3–S4 (not S5). Also, in an S5 state, the Power Button can wake the system, but the Sleep Button cannot.

Although the PCH does not include a specific signal designated as a Sleep Button, one of the GPIO signals can be used to create a "Control Method" Sleep Button. Refer the Advanced Configuration and Power Interface Specification for implementation details.

23.7.7.2 PME# (PCI Power Management Event)

The PME# signal comes from a PCI Express* device to request that the system be restarted. The PME# signal can generate an SMI#, SCI, or optionally a wake event. The event occurs when the PME# signal goes from high to low. No event is caused when it goes from low to high.

There is also an internal PME_B0_STS bit (replaced PME_B0 bit) that will be set by the PCH when any internal device with PCI Power Management capabilities on bus 0 asserts the equivalent of the PME# signal. This is separate from the external PME# signal and can cause the same effect.

23.7.7.3 SYS_RESET# Signal

When the SYS_RESET# pin is detected as active (on signal's falling edge if de-bounce logic is disabled, or after 16 ms if 16ms de-bounce logic is enabled), the PCH attempts to perform a "graceful" reset by entering a host partition reset entry sequence.

Once the reset is asserted, it remains asserted for 5 to 6 ms regardless of whether the SYS_RESET# input remains asserted or not. It cannot occur again until SYS_RESET# has been detected inactive after the de-bounce logic, and the system is back to a full S0 state with PLTRST# inactive.

Note: The normal behavior for a SYS_RESET# assertion is host partition reset without power cycle. However, if bit 3 of the CF9h I/O register is set to '1' then SYS_RESET# will result in a full power-cycle reset.

Note: It is not recommended to use the PCH_PWROK pin for a reset button as it triggers a global power cycle reset.

Note: SYS_RESET# is in the primary power well but it only affects the system when PCH_PWROK is high.

23.7.7.4 THERMTRIP# Signal

If THERMTRIP# goes active, the processor is indicating an overheat condition, and the PCH immediately transitions to an S5 state, driving SLP_S3#, SLP_S4#, SLP_S5# low, and setting the GEN_PMCON_2.PTS bit. The transition will generally look like a power button override.

When a THERMTRIP# event occurs, the PCH will power down immediately without following the normal S0 -> S5 path. The PCH will immediately drive SLP_S3#, SLP_S4#, and SLP_S5# low within 1 us after sampling THERMTRIP# active.

The reason the above is important is as follow: if the processor is running extremely hot and is heating up, it is possible (although very unlikely) that components around it, such as the PCH, are no longer executing cycles properly. Therefore, if THERMTRIP# goes active, and the PCH is relying on various handshakes to perform the power down, the handshakes may not be working, and the system will not power down. Hence the need for PCH to power down immediately without following the normal S0 -> S5 path.

The PCH provides filtering for short low glitches on the THERMTRIP# signal in order to prevent erroneous system shut downs from noise. Glitches shorter than 25 nsec are ignored.

PCH must only honor the THERMTRIP# pin while it is being driven to a valid state by the processor. The THERMTRIP# Valid Point =’0’, implies PCH will start monitoring THERMTRIP# at PLTRST# de-assertion (default). The THERMTRIP# Valid Point =’1’, implies PCH will start monitoring THERMTRIP# at PROCPWRGD assertion. Regardless of the setting, the PCH must stop monitoring THERMTRIP# at PROCPWRGD de-assertion.

Note: A thermal trip event will clear the PWRBTN_STS bit.

23.7.7.5 Sx_Exit_Holdoff#

When S3/S4/S5 is entered and SLP_A# is asserted, Sx_Exit_Holdoff# can be asserted by a platform component to delay resume to S0. SLP_A# de-assertion is an indication of the intent to resume to S0, but this will be delayed so long as Sx_Exit_Holdoff# is asserted. Sx_Exit_Holdoff is ignored outside of an S3/S4/S5 entry sequence with SLP_A# asserted. With the de-assertion of RSMRST# (either from G3->S0 or Deep Sx->S0), this pin is a GPIO input and must be programmed by BIOS to operate as Sx_Exit_Holdoff. When SLP_A# is asserted (or it is de-asserted but Sx_Exit_Holdoff# is asserted), the PCH will not access SPI Flash. How a platform uses this signal is platform specific.

Requirements to support Sx_Exit_Holdoff#:

If the PCH is in G3/Deep Sx or in the process of exiting G3/Deep Sx (RSMRST# is asserted), the EC must not allow RSMRST# to de-assert until the EC completed its flash accesses.

After the PCH has booted up to S0 at least once since the last G3 or Deep Sx exit, the EC can begin monitoring SLP_A# and using the SX_EXIT_HOLDOFF# pin to stop the PCH from accessing flash. When SLP_A# asserts, if the EC intends to access flash, it

will assert SX_EXIT_HOLDOFF#. To cover the case where the PCH is going through a global reset, and not a graceful Sx+CMoff/Sx+CM3PG entry, the EC must monitor the SPI flash CS0# pin for 5ms after SLP_A# assertion before making the determination that it is safe to access flash.

- If no flash activity is seen within this 5ms window, the EC can begin accessing flash. Once its flash accesses are complete, the EC de-asserts (drives to '1') SX_EXIT_HOLDOFF# to allow the PCH to access flash.
- If flash activity is seen within this 5ms window, the PCH has gone through a global reset. And so the EC must wait until the PCH reaches S0 again before re-attempting the holdoff flow.

Note: When eSPI is enabled, SX_EXIT_HOLDOFF# functionality is not available, and assertion of the signal will not impact Sx exit flows.

23.7.8 System Power Supplies, Planes, and Signals

23.7.8.1 Power Plane Control

The SLP_S3# output signal can be used to cut power to the system core supply, since it only goes active for the Suspend-to-RAM state (typically mapped to ACPI S3). Power must be maintained to the PCH primary well, and to any other circuits that need to generate Wake signals from the Suspend-to-RAM state. During S3 (Suspend-to-RAM) all signals attached to powered down planes will be tri-stated or driven low, unless they are pulled using a Pull-up resistor.

Cutting power to the system core supply may be done using the power supply or by external FETs on the motherboard.

The SLP_S4# or SLP_S5# output signal can be used to cut power to the system core supply, as well as power to the system memory, since the context of the system is saved on the disk. Cutting power to the memory may be done using the power supply, or by external FETs on the motherboard.

SLP_S5# output signal can be used to cut power to the system core supply.

SLP_LAN# output signal can be used to cut power to the external Intel GbE PHY device.

23.7.8.2 SLP_S4# and Suspend-to-RAM Sequencing

The system memory suspend voltage regulator is controlled by the Glue logic. The SLP_S4# signal should be used to remove power to system memory rather than the SLP_S5# signal. The SLP_S4# logic in the PCH provides a mechanism to fully cycle the power to the DRAM and/or detect if the power is not cycled for a minimum time.

Note: To use the minimum DRAM power-down feature that is enabled by the SLP_S4# Assertion Stretch Enable bit (D31:F0:A4h Bit 3), the DRAM power must be controlled by the SLP_S4# signal.

23.7.8.3 PCH_PWROK Signal

When asserted, PCH_PWROK is an indication to the PCH that its core well power rails are powered and stable. PCH_PWROK can be driven asynchronously. When PCH_PWROK is low, the PCH asynchronously asserts PLTRST#. PCH_PWROK must not glitch, even if RSMRST# is low.

It is required that the power associated with PCIe* have been valid for 99 ms prior to PCH_PWROK assertion in order to comply with the 100 ms PCIe* 2.0 specification on PLTRST# de-assertion.

Note: SYS_RESET# is recommended for implementing the system reset button. This saves external logic that is needed if the PCH_PWROK input is used. Additionally, it allows for better handling of the SMBus and processor resets and avoids improperly reporting power failures.

23.7.8.4 BATLOW# (Battery Low)

The BATLOW# input can inhibit waking from S3, S4, S5 and Deep Sx states if there is not sufficient power. It also causes an SMI if the system is already in an S0 state.

23.7.8.5 SYS_PWROK

SYS_PWROK is a generic power good input to the PCH that is driven and utilized in a platform-specific manner. While PCH_PWROK always indicates that power to the CORE well devices on the board are stable, SYS_PWROK is used to inform the PCH that power is stable to some other system components and the system is ready to start the exit from reset.

23.7.8.6 DRAM_RESET#

The DRAM_RESET# output pin is sent to the DRAM as an indication of DRAM power stability. This pin is routed to the DDR_RESET# input pin on the DRAM.

Note: PCH does not directly know when the DRAM power is valid. However, the PCH assumes that asserting the SLP_S4# pin will result in DRAM power loss.

23.7.8.7 SLP_LAN# Pin Behavior

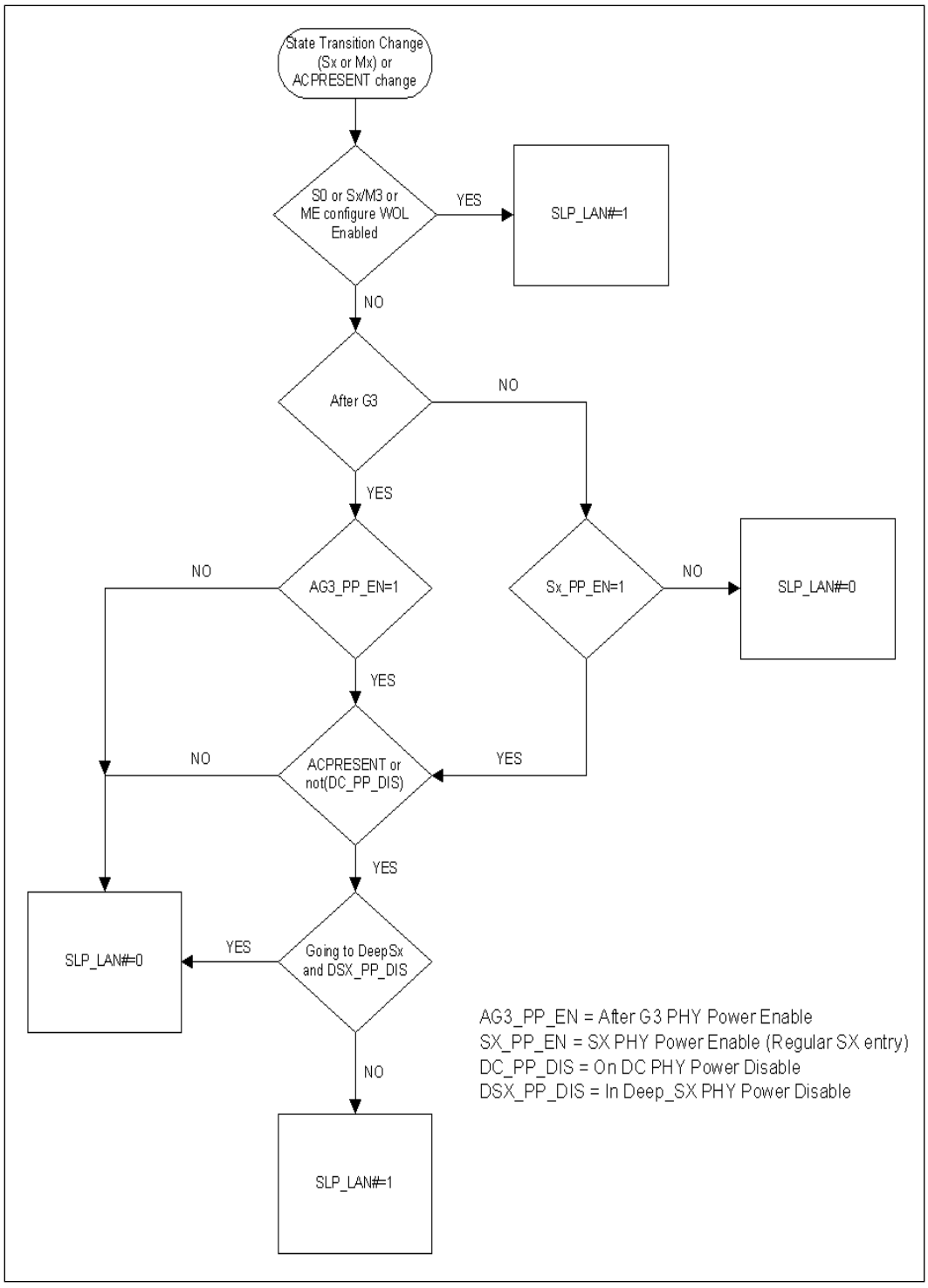
The PCH controls the voltage rails into the external LAN PHY using the SLP_LAN# pin.

- The LAN PHY is always powered when the Host and Intel® CSME systems are running.
  - SLP_LAN#='1' whenever SLP_S3#='1' or SLP_A#='1'.
- If the LAN PHY is required by Intel® CSME in Sx/M-Off or Deep Sx, Intel® CSME must configure SLP_LAN#='1' irrespective of the power source and the destination power state. Intel® CSME must be powered at least once after G3 to configure this.
- If the LAN PHY is required after a G3 transition, the host BIOS must set AG3_PP_EN.
- If the LAN PHY is required in Sx/M-Off, the host BIOS must set SX_PP_EN.
- If the LAN PHY is required in Deep Sx, the host BIOS must keep DSX_PP_DIS cleared.
- If the LAN PHY is not required if the source of power is battery, the host BIOS must set DC_PP_DIS.

Note: Intel® CSME configuration of SLP_LAN# in Sx/M-Off and Deep Sx is dependent on Intel® CSME power policy configuration.

The flow chart below shows how a decision is made to drive SLP_LAN# every time its policy needs to be evaluated.

Figure 23-1. Conceptual Diagram of SLP_LAN#



23.7.8.8 SLP_WLAN# Pin Behavior

The PCH controls the voltage rails into the external wireless LAN PHY using the SLP_WLAN# pin.

- The wireless LAN PHY is always powered when the Host is running.
  - SLP_WLAN#='1' whenever SLP_S3#='1'.
- If Wake on Wireless LAN (WoWLAN) is required from S3/S4/S5 states, the host BIOS must set HOST_WLAN_PP_EN.
- If WoWLAN is required from Deep Sx, the host BIOS must set DSX_WLAN_PP_EN.
- If Intel® CSME has access to the Wireless LAN device:
  - The Wireless LAN device must always be powered as long as Intel® CSME is powered. SLP_WLAN#='1' whenever SLP_A#='1'.
  - If Wake on Wireless LAN (WoWLAN) is required from M-Off state, Intel® CSME will configure SLP_WLAN#='1' in Sx/M-Off.

Intel® CSME configuration of SLP_WLAN# in Sx/M-Off is dependent on Intel® CSME power policy configuration.

23.7.8.9 SUSPWRDNACK/SUSWARN# Steady State Pin Behavior

Table 23-12 summarizes SUSPWRDNACK/SUSWARN# pin behavior.

Table 23-12. SUSPWRDNACK/SUSWARN#/GPP_C6 Pin Behavior

Pin	Deep Sx (Supported /Not-Supported)	GPP_C6 Input/Output (Determine by GP_IO_SEL bit)	Pin Value in S0	Pin Value in Sx/M-Off	Pin Value in Sx/M3	Pin Value in Deep Sx
SUSPWRDNACK	Not Supported	Native	0	Depends on Intel® CSME power package and power source (Note 1)	0	OFF
SUSWARN#	Supported	Native	1	1 (Note 2)	1	OFF
GPP_C6	Do not Care	IN	High-Z	High-Z	High-Z	OFF
	Do not Care	OUT	Depends on GPP_C6 output data value	Depends on GPP_C6 output data value	Depends on GPP_C6 output data value	OFF

Notes:
1. PCH will drive SPDA pin based on Intel® CSME power policy configuration.
2. If entering Deep Sx, pin will assert and become undriven ("Off") when suspend well drops upon Deep Sx entry.

Table 23-13. SUSPWRDNACK During Reset

Reset Type (Note)	SPDA Value
power-cycle Reset	0
Global Reset	0
Straight to S5	PCH initially drive '0' and then drive per Intel® CSME power policy configuration.

Note: Refer Table 23-16

23.7.8.10 RTCRST# and SRTCRST#

RTCRST# is used to reset PCH registers in the RTC Well to their default value. If a jumper is used on this pin, it should only be pulled low when system is in the G3 state and then replaced to the default jumper position. Upon booting, BIOS should recognize that RTCRST# was asserted and clear internal PCH registers accordingly. It is imperative that this signal not be pulled low in the S0 to S5 states.

SRTCRST# is used to reset portions of the Intel Management Engine and should not be connected to a jumper or button on the platform. The only time this signal gets asserted (driven low in combination with RTCRST#) should be when the coin cell battery is removed or not installed and the platform is in the G3 state. Pulling this signal low independently (without RTCRST# also being driven low) may cause the platform to enter an indeterminate state. Similar to RTCRST#, it is imperative that SRTCRST# not be pulled low in the S0 to S5 states.

23.7.9 Reset Behavior

When a reset is triggered, the PCH will send a warning message to the processor to allow the processor to attempt to complete any outstanding memory cycles and put memory into a safe state before the platform is reset. When the processor is ready, it will send an acknowledge message to the PCH. Once the message is received the PCH asserts PLTRST#.

The PCH does not require an acknowledge message from the processor to trigger PLTRST#. A global reset will occur after 4 seconds if an acknowledge from the processor is not received.

When the PCH causes a reset by asserting PLTRST#, its output signals will go to their reset states.

A reset in which the host platform is reset and PLTRST# is asserted is called a Host Reset or Host Partition Reset. Depending on the trigger a host reset may also result in power cycling refer Table 23-16 for details. If a host reset is triggered and the PCH times out before receiving an acknowledge message from the processor a Global Reset with power-cycle will occur.

A reset in which the host and Intel® CSME partitions of the platform are reset is called a Global Reset. During a Global Reset, all PCH functionality is reset except RTC Power Well backed information and Suspend well status, configuration, and functional logic for controlling and reporting the reset. Intel® CSME and Host power back up after the power-cycle period.

Straight to S5 is another reset type where all power wells that are controlled by the SLP_S3#, SLP_S4#, and SLP_A# pins, as well as SLP_S5# and SLP_LAN# (if pins are not configured as GPIOs), are turned off. All PCH functionality is reset except RTC Power Well backed information and Suspend well status, configuration, and functional logic for controlling and reporting the reset. The host stays there until a valid wake event occurs.

Table 23-16 shows the various reset triggers.

Table 23-16. Causes of Host and Global Resets (Sheet 1 of 2)

Trigger	Host Reset Without Power Cycle[1]	Host Reset With Power Cycle[2]	Global Reset With Power Cycle[3]	Straight to S5[6] (Host Stays There)
Write of 0Eh to CF9h (RST_CNT Register) when CF9h when Global Reset Bit=0b	No	Yes	No (Note 4)	
Write of 06h to CF9h (RST_CNT Register) when CF9h when Global Reset Bit=0b	Yes	No	No (Note 4)	
Write of 06h or 0Eh to CF9h (RST_CNT Register) when CF9h when Global Reset Bit=1b	No	No	Yes	
SYS_RESET# Asserted and CF9h (RST_CNT Register) Bit 3 = 0	Yes	No	No (Note 4)	
SYS_RESET# Asserted and CF9h (RST_CNT Register) Bit 3 = 1	No	Yes	No (Note 4)	
SMBus Slave Message received for Reset with Power-Cycle	No	Yes	No (Note 4)	
SMBus Slave Message received for Reset without Power-Cycle	Yes	No	No (Note 4)	
SMBus Slave Message received for unconditional Power Down	No	No	No	Yes
TCO Watchdog Timer reaches zero two times	Yes	No	No (Note 4)	
Power Failure: PCH_PWROK signal goes inactive in S0 or DSW_PWROK drops	No	No	Yes	
SYS_PWROK Failure: SYS_PWROK signal goes inactive in S0	No	No	Yes	
Processor Thermal Trip (THERMTRIP#) causes transition to S5 and reset asserts	No	No	No	Yes
PCH internal thermal sensors signals a catastrophic temperature condition	No	No	No	Yes
Power Button 4 second override causes transition to S5 and reset asserts	No	No	No	Yes
Special shutdown cycle from processor causes CF9h-like PLTRST# and CF9h Global Reset Bit = 1	No	No	Yes	
Special shutdown cycle from processor causes CF9h-like PLTRST# and CF9h Global Reset Bit = 0 and CF9h (RST_CNT Register) Bit 3 = 1	No	Yes	No (Note 4)	
Special shutdown cycle from processor causes CF9h-like PLTRST# and CF9h Global Reset Bit = 0 and CF9h (RST_CNT Register) Bit 3 = 0	Yes	No	No (Note 4)	
Intel® Management Engine Triggered Host Reset without Power-Cycle	Yes	No	No (Note 4)	
Intel® Management Engine Triggered Host Reset with Power-Cycle	No	Yes	No (Note 4)	
Intel® Management Engine Triggered Power Button Override	No	No	No	Yes
Intel® Management Engine Watchdog Timer Timeout	No	No	No (Note 8)	Yes

Table 23-16. Causes of Host and Global Resets (Sheet 2 of 2)

Trigger	Host Reset Without Power Cycle[1]	Host Reset With Power Cycle[2]	Global Reset With Power Cycle[3]	Straight to S5[6] (Host Stays There)
Intel® Management Engine Triggered Global Reset	No	No	Yes	
Intel® Management Engine Triggered Host Reset with power down (host stays there)	No	Yes (Note 5)	No (Note 4)	
PLTRST# Entry Timeout (Note 7)	No	No	Yes	
PROCPWRGD Stuck Low	No	No	Yes	
Power Management Watchdog Timer	No	No	No (Note 8)	Yes
Intel® Management Engine Hardware Uncorrectable Error	No	No	No (Note 8)	Yes

Notes:
1. The PCH drops this type of reset request if received while the system is in S3/S4/S5.
2. PCH does not drop this type of reset request if received while system is in a software-entered S3/S4/S5 state. However, the PCH will perform the reset without executing the RESET_WARN protocol in these states.
3. The PCH does not send warning message to processor, reset occurs without delay.
4. Trigger will result in Global Reset with Power-Cycle if the acknowledge message is not received by the PCH.
5. The PCH waits for enabled wake event to complete reset.
6. Upon entry to S5, if Deep Sx is enabled and conditions are met per Section 23.7.6.6, the system will transition to Deep Sx.
7. PLTRST# Entry Timeout is automatically initiated if the hardware detects that the PLTRST# sequence has not been completed within 4 seconds of being started.
8. Trigger will result in Global Reset with Power-Cycle if AGR_LS_EN=1 and Global Reset occurred while the current or destination state was S0.

23.8 Fully Integrated Voltage Regulator (FIVR)

The PCH integrates multiple voltage rails onto the PCH in order to reduce BOM costs for the platform. There will be 3 FIVRs integrated on the ICP-LP (VCCIO, VCC_VNNEXT_1P05, VCC_V1P05EXT_1P05). Each FIVR is able to control a specific voltage rail. Below table shows the PCH platform voltage rails.

Table 23-17. PCH Platform Voltage Rails

Rail	Voltage	Description
VCCIN_AUX	1.8 V or 1.65 V	FIVR Input rail (active). 1.8 V is boot voltage for Y sku.
VCCPRIM_1P8	1.8 V	GPIO supply voltage
VCCDSW_3P3	3.3 V	3.3V Deep sleep well supply
VCCPRIM_3P3	3.3 V	Primary 3.3V supply
VCCRTC	3 V	RTC Supply (3.0-3.3)
VCC_VNNEXT_1P05 (Optional)	1.05 V	Used for FIVR PRIM_CORE bypass mode during Sx: 1.05V
VCC_V1P05EXT_1P05 (Optional)	1.05 V	Used for FIVR PCH IO bypass mode during Sx: 1.05V

23.8.1 VCCIN_AUX

VCCIN_AUX is the external input rail to FIVR for generating the internal voltage rails and this rail is shared between CPU and PCH. During deep S0ix, and Sx states, VCCIN_AUX may be shutdown. In general, placing controllers and devices into lower power modes such as D3 or device disconnected will result in a reduced total current requirement and aid in achieving VCCIN_AUX shutdown. This will be done by driving the CORE_VID values to '00. Vccin_AUX powergood during initial reset is tied into the RSMRST# signal, requiring that the FIVR input voltage rail is stable in the same window as the other SLP_SUS# rails.

VCCIN_AUX is controlled by CORE_VID[1:0].

Table 23-18. VCCIN_AUX Control

CORE_VID1	CORE_VID0	VCCIN_AUX	Comments
0	0	0 V	Typically used in S0ix states
0	1	1.1 V	Retention FIVR voltage
1	0	1.65 V	Low current mode voltage
1	1	1.8 V	High current mode voltage (boot voltage)

23.8.2 External Bypass Rails

The VCC_VNNEXT_1P05 and VCC_V1P05EXT_1P05 rails can also have an input from a separate external voltage rail. These rails are always on and must come up after the V1p8A rail has been brought up. Note that there is no feedback that this rail is valid.

§ §

24 Real Time Clock (RTC)

24.1 Acronyms

Acronyms	Description
BCD	Binary Coded Decimal
CMOS	Complementary Metal Oxide Semiconductor. A manufacturing process used to produce electronics circuits, but in reference to RTC is used interchangeably as the RTC's RAM i.e. clearing CMOS meaning to clear RTC RAM.
ESR	Equivalent Series Resistance. Resistive element in a circuit such as a clock crystal.
GPI	General Purpose Input
PPM	Parts Per Million. Used to provide crystal accuracy or as a frequency variation indicator.
RAM	Random Access Memory

24.2 Feature Overview

The PCH contains a real-time clock functionally compatible with the Motorola MC146818B. The real-time clock has 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 as long as the RTC power well is powered. The RTC operates on a 32.768 kHz oscillating source and a 3V battery or system battery if configured by design as the source.

The RTC also supports two lockable memory ranges. By setting bits in the configuration space, two 8-byte ranges can be locked to read and write accesses. This prevents unauthorized reading of passwords or other system security information.

The RTC also supports a date alarm that allows for scheduling a wake up event up to month in advance.

24.3 Signal Description

Name	Type	Description
RTCX1	I	**Crystal Input 1:** This signal is connected to the 32.768 kHz crystal (max 50K ohm ESR). If no external crystal is used, then RTCX1 can be driven with the desired clock rate. Maximum voltage allowed on this pin is 1.5V.
RTCX2	O	**Crystal Input 2:** This signal is connected to the 32.768 kHz crystal (max 50K ohm ESR). If no external crystal is used, then RTCX2 must be left floating.
RTCRST#	I	**RTC Reset:** When asserted, this signal resets register bits in the RTC well. ***Notes:*** 1. Unless CMOS is being cleared (only to be done in the G3 power state) with a jumper, the RTCRST# input must always be high when all other RTC power planes are on. 2. In the case where the RTC battery is dead or missing on the platform, the RTCRST# pin must rise before the DSW_PWROK pin.

Name	Type	Description
SRTCRST#	I	**Secondary RTC Reset:** This signal resets the manageability register bits in the RTC well when the RTC battery is removed. ***Notes:*** 1. The SRTCRST# input must always be high when all other RTC power planes are on. 2. In the case where the RTC battery is dead or missing on the platform, the SRTCRST# pin must rise before the DSW_PWROK pin. 3. SRTCRST# and RTCRST# should not be shorted together.

24.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S3/S4/S5	Deep Sx
RTCRST#	RTC	Undriven	Undriven	Undriven	Undriven
SRTCRST#	RTC	Undriven	Undriven	Undriven	Undriven
Notes: 1. Reset reference for primary well pins is RSMRST#.					

§ §

25 Serial ATA (SATA)

25.1 Acronyms

Acronyms	Description
AHCI	Advanced Host Controller Interface
DMA	Direct Memory Access
DEVSLP	Device Sleep
IDE	Integrated Drive Electronics
RAID	Redundant Array of Independent Disks
SATA	Serial Advanced Technology Attachment

25.2 References

Specification	Location
Serial ATA Specification, Revision 3.2	https://www.sata-io.org
Serial ATA II: Extensions to Serial ATA 1.0, Revision 1.0	https://www.sata-io.org
Serial ATA II Cables and Connectors Volume 2 Gold	https://www.sata-io.org
Advanced Host Controller Interface Specification	http://www.intel.com/content/www/us/en/io/serial-ata/ahci.html

25.3 Overview

The PCH SATA controller support two modes of operation, AHCI mode using memory space and RAID mode. The PCH SATA controller no longer supports IDE legacy mode using I/O space. Therefore, AHCI software is required. The PCH SATA controller supports the Serial ATA Specification, Revision 3.2.

Note: Not all functions and capabilities may be available on all SKUs. Refer to PCH-U/Y I/O Capabilities table and PCH-U/Y SKUs table for details on feature availability.

25.4 Signals Description

Table 25-1. Signals Description (Sheet 1 of 3)

Name	Type	Description
DEVSLP0/ GPP_E4	OD	**Serial ATA Port [0] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state. Design Constraint: No external Pull-up or Pull-down termination required when used as DEVSLP. ***Note:*** This pin can be mapped to SATA Port 0.

Table 25-1. Signals Description (Sheet 2 of 3)

Name	Type	Description
DEVSLP1/ GPP_E5	OD	**Serial ATA Port [1] Device Sleep:** This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state. Design Constraint: No external Pull-up or Pull-down termination required when used as DEVSLP. ***Note:*** This pin can be mapped to SATA Port 1.
DEVSLP2/ GPP_A11	OD	**Serial ATA Port [2] Device Sleep:** This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state. Design Constraint: No external Pull-up or Pull-down termination required when used as DEVSLP. ***Note:*** This pin can be mapped to SATA Port 2.
SATA0_TXP/ PCIE11_TXP **SATA0_TXN**/ PCIE11_TXN	O	**Serial ATA Differential Transmit Pair 0:** These outbound SATA Port 0 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 11 signals.
SATA0_RXP/ PCIE11_RXP **SATA0_RXN**/ PCIE11_RXN	I	**Serial ATA Differential Receive Pair 0:** These inbound SATA Port 0 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 11 signals.
SATA1A_TXP/ PCIE12_TXP **SATA1A_TXN**/ PCIE12_TXN	O	**Serial ATA Differential Transmit Pair 1 [First Instance]:** These outbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 12.
SATA1A_RXP/ PCIE12_RXP **SATA1A_RXN**/ PCIE12_RXN	I	**Serial ATA Differential Receive Pair 1 [First Instance]:** These inbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 12.
SATA1B_TXP/ PCIE15_TXP **SATA1B_TXN**/ PCIE15_TXN	O	**Serial ATA Differential Transmit Pair 1 [Second Instance]:** These outbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 15 signals.
SATA1B_RXP/ PCIE15_RXP **SATA1B_RXN/** PCIE15_RXN	I	**Serial ATA Differential Receive Pair 1 [Second Instance]:** These inbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 15 signals.
SATA2_TXP/ PCIE16_TXP **SATA2_TXN/** PCIE16_TXN	O	**Serial ATA Differential Transmit Pair 2 (PCH-U Only):** These outbound SATA Port 2 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 16 signals.
SATA2_RXP/ PCIE16_RXP **SATA2_RXN**/ PCIE16_RXN /	I	**Serial ATA Differential Receive Pair 2 (PCH-U Only):** These inbound SATA Port 2 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 16 signals.
SATAGP0/ GPP_E0 / SATAXPCIE0	I	**Serial ATA Port [0] General Purpose Inputs:** When configured as SATAGP0, this is an input pin that is used as an interlock switch status indicator for SATA Port 0. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open. ***Note:*** The default use of this pin is GPP_E0. Pin defaults to Native mode as SATAXPCIE0 depends on soft-strap.
SATAGP1/ GPP_A12 / SATAXPCIE1	I	**Serial ATA Port [1] General Purpose Inputs:** When configured as SATAGP1, this is an input pin that is used as an interlock switch status indicator for SATA Port 1. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open. ***Note:*** This default use of this pin is GPP_E1. Pin defaults to Native mode as SATAXPCIE1 depends on soft-strap.

Table 25-1. Signals Description (Sheet 3 of 3)

Name	Type	Description
SATAGP2/ GPP_A13 / SATAXPCIE2	I	**Serial ATA Port [2] General Purpose Inputs:** When configured as SATAGP2, this is an input pin that is used as an interlock switch status indicator for SATA Port 2. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open. ***Note:*** The default use of this pin is GPP_E2. Pin defaults to Native mode as SATAXPCIE2 depends on soft-strap.
SATALED#/ GPP_E8 / SPI1_CS1#	OD O	**Serial ATA LED:** This signal is an open-drain 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. ***Note:*** An external Pull-up resistor to VCC3_3 is required.

25.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor type	Notes
SATAXPCIE[2:0]	Internal pull-up	Internal Pull-Up Resistors are 15k-40k unless specified.

25.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[3]	Immediately after Reset[3]	S3/S4/S5	Deep Sx
SATA0_TXP/N, SATA0_RXP/N	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SATA1A_TXP/N, SATA1A_RXP/N	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SATA1B_TXP/N, SATA1B_RXP/N	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SATA2_TXP/N, SATA2_RXP/N	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SATALED#/GPP_E81	Primary	Undriven	Undriven	Undriven	OFF
DEVSLP[2:0]/ GPP_E[6:4][1]	Primary	Undriven	Undriven	Driven Low	OFF
SATAGP[2:0]/ GPP_E[2:0][2]	Primary	Undriven	Undriven	Undriven	OFF
SATAXPCIE[2:0][2]	Primary	Internal Pull-up	Internal Pull-up	Undriven	OFF

Note:
1. Pin defaults to GPIO mode. The pin state during and immediately after reset follows default GPIO mode pin state. The pin state for S0 to Deep Sx reflects assumption that GPIO Use Select register was programmed to native mode functionality. If GPIO Use Select register is programmed to GPIO mode, refer to Multiplexed GPIO (Defaults to GPIO Mode) section for the respective pin states in S0 to Deep Sx.
2. Pin defaults to Native mode as SATAXPCIEx depends on soft-strap.
3. Reset reference for primary well pins is RSMRST#.

25.7 Functional Description

The PCH SATA host controller (D23:F0) supports AHCI or RAID mode.

The PCH SATA controller does not support legacy IDE mode or combination mode.

The PCH SATA controller interacts with an attached mass storage device through a register interface that is compatible with an SATA AHCI/RAID host adapter. The host software follows existing standards and conventions when accessing the register interface and follows standard command protocol conventions.

25.7.1 SATA 6 Gb/s Support

The PCH SATA controller is SATA 6 Gb/s capable and supports 6 Gb/s transfers with all capable SATA devices. The PCH SATA controller also supports SATA 3 Gb/s and 1.5 Gb/s transfer capabilities.

25.7.2 SATA Feature Support

The PCH SATA controller is capable of supporting all AHCI 1.3 and AHCI 1.3.1, refer to the Intel web site on Advanced Host Controller Interface Specification for current specification status: http://www.intel.com/content/www/us/en/io/serial-ata/ahci.html.

For capability details, refer to PCH SATA controller register (D23:F0:Offset 00h CAP, and AHCI BAR PxCMD Offset 18h).

The PCH SATA controller does **not** support:

- Port Multiplier
- FIS Based Switching
- Command Based Switching
- IDE mode or combination mode
- Cold Presence Detect
- Function Level Reset (FLR)

25.7.3 Hot-Plug Operation

The PCH SATA controller supports Hot-Plug Surprise removal and Insertion Notification. An internal SATA port with a Mechanical Presence Switch can support PARTIAL and SLUMBER with Hot-Plug Enabled. Software can take advantage of power savings in the low power states while enabling Hot-Plug operation. Refer to Chapter 7 of the AHCI specification for details.

25.7.4 Intel® Rapid Storage Technology (Intel® RST)

The PCH SATA controller provides support for Intel® Rapid Storage Technology, providing both AHCI and integrated RAID functionality. The RAID capability provides high-performance/data-redundancy RAID 0/1 functionality on up to two ports for PCH-Y and RAID 0/1/5 functionality on up to three ports for PCH-U of the PCH SATA controller. Matrix RAID support is provided to allow multiple RAID levels to be combined on a single set of hard drives, such as RAID 0 and RAID 1 on two disks. Other RAID features include hot spare support, SMART alerting, and RAID 0 auto replace. Software

components include an Option ROM and UEFI Driver for pre-boot configuration and boot functionality, a Microsoft* Windows* compatible driver, and a user interface for configuration and management of the RAID capability of PCH SATA controller.

Note: Not all functions and capabilities may be available on all SKUs. Refer to PCH-U/Y I/O Capabilities table and PCH-U/Y SKUs table for details on feature availability.

25.7.4.1 Intel® Rapid Storage Technology (Intel® RST) Configuration

Intel® RST offers several diverse options for RAID (redundant array of independent disks) to meet the needs of the end user. AHCI support provides higher performance and alleviates disk bottlenecks by taking advantage of the independent DMA engines that each SATA port offers in the PCH SATA controller.

- RAID Level 0 performance scaling up to 6 drives, enabling higher throughput for data intensive applications such as video editing.
- Data redundancy is offered through RAID Level 1, which performs mirroring.
- RAID Level 5 provides highly efficient storage while maintaining fault-tolerance on 3 or more drives. By striping parity, and rotating it across all disks, fault tolerance of any single drive is achieved while only consuming 1 drive worth of capacity. That is, a 3-drive RAID 5 has the capacity of 2 drives, or a 4-drive RAID 5 has the capacity of 3 drives. RAID 5 has high read transaction rates, with a medium write rate. RAID 5 is well suited for applications that require high amounts of storage while maintaining fault tolerance.

By using the PCH's built-in Intel® Rapid Storage Technology, there is no loss of additional PCIe*/system resources or add-in card slot/motherboard space footprint used compared to when a discrete RAID controller is implemented. Intel® Rapid Storage Technology functionality requires the following items:

1. PCH SKU enabled for Intel® Rapid Storage Technology.
   Note: Not all functions and capabilities may be available on all SKUs. Refer to PCH-U/Y I/O Capabilities table and PCH-U/Y SKUs table for details on feature availability.
2. Intel® Rapid Storage Technology RAID Option ROM or UEFI Driver must be on the platform.
3. Intel® Rapid Storage Technology drivers, most recent revision.
4. At least two SATA hard disk drives (minimum depends on RAID configuration).

Intel® Rapid Storage Technology is not available in the following configurations:

1. The SATA controller is programmed in RAID mode, but the AIE bit (D23:F0:Offset 9Ch bit 7) is set to 1.

25.7.4.2 Intel® Rapid Storage Technology (Intel® RST) RAID Option ROM

The Intel® Rapid Storage Technology RAID Option ROM is a standard PnP Option ROM that is easily integrated into any System BIOS. When in place, it provides the following three primary functions:

- Provides a text mode user interface that allows the user to manage the RAID configuration on the system in a pre-operating system environment. Its feature set is kept simple to keep size to a minimum, but allows the user to create and delete RAID volumes and select recovery options when problems occur.
- Provides boot support when using a RAID volume as a boot disk. It does this by providing Int13 services when a RAID volume needs to be accessed by MS-DOS

applications (such as NTLDR) and by exporting the RAID volumes to the System BIOS for selection in the boot order.

- At each boot up, provides the user with a status of the RAID volumes and the option to enter the user interface by pressing CTRL-I.

25.7.5 Power Management Operation

Power management of the PCH SATA controller and ports will cover operations of the host controller and the SATA link.

25.7.5.1 Power State Mappings

The D0 PCI Power Management (PM) state for device is supported by the PCH SATA controller.

SATA devices may also have multiple power states. SATA adopted 3 main power states from parallel ATA. The three device states are supported through ACPI. They are:

- **D0** – Device is working and instantly available.
- **D1** – Device enters when it receives a STANDBY IMMEDIATE command. Exit latency from this state is in seconds.
- **D3** – From the SATA device's 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 will reset the device and then cut its power.

Each of these device states are subsets of the host controller's D0 state.

Finally, the SATA specification 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 in active state.
- **Partial** – PHY logic is powered up, and in a reduced power state. The link PM exit latency to active state maximum is 10 ns.
- **Slumber** – PHY logic is powered up, and in a reduced power state. The link PM exit latency to active state maximum is 10 ms.
- **Devslp** – PHY logic is powered down. The link PM exit latency from this state to active state maximum is 20 ms, unless otherwise specified by DETO in Identify Device Data Log page 08h (Refer SATA Rev3.2 Gold specification).

Since these states have much lower exit latency than the ACPI D1 and D3 states, the SATA controller specification defines these states as sub-states of the device D0 state.

25.7.5.2 Power State Transitions

25.7.5.2.1 Partial and Slumber State Entry/Exit

The partial and slumber states save interface power when the interface is idle. It would be most analogous to CLKRUN# (in power savings, not in mechanism), where the interface can have 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 will be ACKed.

When an operation is performed to the SATA controller such that it 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 COMWAKE to bring the link back online. Similarly, the SATA device must perform the same COMWAKE action.

Note: SATA devices shall not attempt to wake the link using COMWAKE/COMINIT when no commands are outstanding and the interface is in Slumber.

25.7.5.2.2 Devslp State Entry/Exit

Device Sleep (DEVSLP) is a host-controlled SATA interface power state. To support a hardware autonomous approach that is software agnostic Intel is recommending that BIOS configure the AHCI controller and the device to enable Device Sleep. This allows the AHCI controller and associated device to automatically enter and exit Device Sleep without the involvement of OS software.

To enter Device Sleep the link must first be in Slumber. By enabling HIPM (with Slumber) or DIPM on a Slumber capable device, the device/host link may enter the DevSleep Interface Power state.

The device must be DevSleep capable. Device Sleep is only entered when the link is in slumber, therefore when exiting the Device Sleep state, the device must resume with the COMWAKE out-of-band signal (and not the COMINIT out-of-band signal). Assuming Device Sleep was asserted when the link was in slumber, the device is expected to exit DEVSLP to the DR_Slumber state. Devices that do not support this feature will not be able to take advantage of the hardware automated entry to Device Sleep that is part of the AHCI 1.3.1 specification and supported by Intel platforms.

25.7.5.2.3 Device D1 and D3 States

These states are entered after some period of time when software has determined that no commands will be 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.

25.7.5.2.4 Host Controller $D3_{HOT}$ State

After the interface and device have been put into a low power state, the SATA host controller may be put into a low power state. This is performed using the PCI power management registers in configuration space. There are two very important aspects to Note 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 will result in master abort.
2. When the power state is D3, no interrupts may be generated, even if they are enabled. If an interrupt status bit is pending when the controller transitions to D0, an interrupt may be generated.

When the controller is put into D3, it is assumed that software has properly shut down the device and disabled the ports. Therefore, there is no need to sustain any values on the port wires. The interface will be treated as if no device is present on the cable, and power will be minimized.

When returning from a D3 state, an internal reset will not be performed.

25.7.5.3 Low Power Platform Consideration

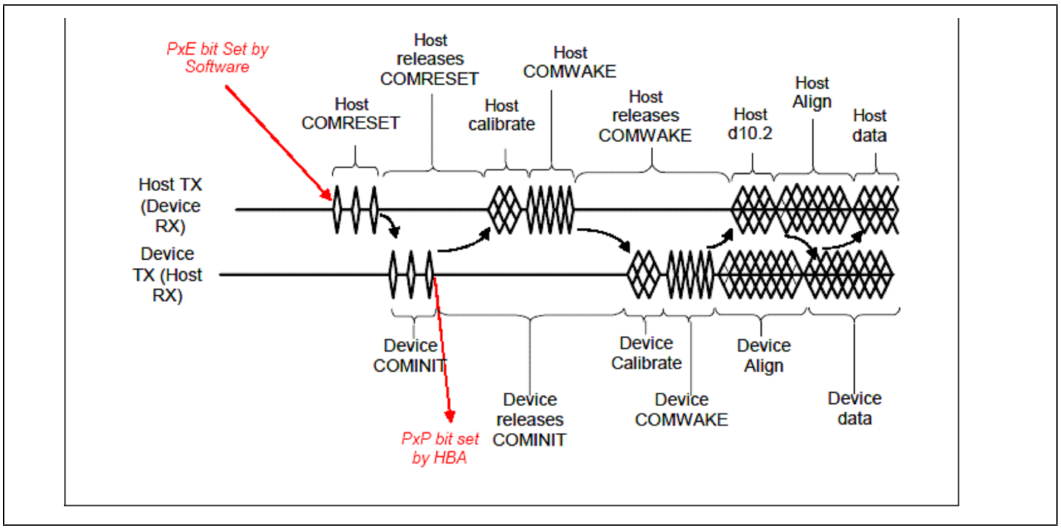
When low power feature is enabled, the Intel SATA controller may power off PLLs or OOB detection circuitry while in the Slumber link power state. As a result, a device initiated wake may not be recognized by the host. For example, when the low power feature is enabled it can prevent a Zero Power ODD (ZPODD) device from successfully communicating with the host on media insertion.

The SATA MPHY Dynamic Power Gating (PHYDPGEPx) can be enabled/disabled for each SATA ports. Refer to SATA SIR Index 50h (for PCH-U/Y) for the PHYDPGEPx register details.

25.7.6 SATA Device Presence

The flow used to indicate SATA device presence is shown in Figure 25-1. The 'PxE' bit refers to PCS.P[2:0]E bits, depending on the port being checked and the 'PxP' bits refer to the PCS.P[2:0]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, software can check to view 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 it can disable the port and check again later. If a port remains enabled, software can periodically poll PCS.PxP to view if a new device is connected.

Figure 25-1. Flow for Port Enable/Device Present Bits



25.7.7 SATA LED

The SATALED# output is driven whenever the BSY bit is set in any SATA port. The SATALED# is an active-low open-drain output. When SATALED# is low, the LED should be active. When SATALED# is high, the LED should be inactive.

25.7.8 Advanced Host Controller Interface (AHCI) Operation

The PCH SATA controller provides hardware support for Advanced Host Controller Interface (AHCI), a standardized programming interface for SATA host controllers developed through a joint industry effort. Platforms supporting AHCI may take advantage of performance features such as port independent DMA Engines—each device is treated as a master—and hardware-assisted native command queuing.

AHCI defines transactions between the SATA controller and software and enables advanced performance and usability with SATA. Platforms supporting AHCI may 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 also provides usability enhancements such as hot-plug and advanced power management. AHCI requires appropriate software support (such as, an AHCI driver) and for some features, hardware support in the SATA device or additional platform hardware. Visit the Intel web site for current information on the AHCI specification.

The PCH SATA controller supports all of the mandatory features of the *Serial ATA Advanced Host Controller Interface Specification*, Revision 1.3.1 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 may be 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 should be disabled for the associated port. Refer Section 7.3.1 of the AHCI Specification for more information.

§ §

26 System Management Interface and SMLink

26.1 Acronyms

Acronyms	Description
BMC	Baseboard Management Controller
EC	Embedded Controller

26.2 Feature Overview

The PCH provides two SMLink interfaces, SMLink0 and SMLink1. The interfaces are intended for system management and are controlled by the Intel® CSME. Refer Section 5 for more detail.

26.3 Signal Description

Name	Type	Description
INTRUDER#	I	**Intruder Detect:** This signal can be used ONLY as SPI voltage select strap.
SML0DATA/ GPP_C4	I/OD	**System Management Link 0 Data:** SMBus link to external PHY. External Pull-up resistor required.
SML0CLK /GPP_C3	I/OD	**System Management Link 0 Clock** External Pull-up resistor required.
SML0ALERT# / GPP_C5	I/OD	**System Management 0 Alert:** Alert for the SMBus controller to optional Embedded Controller or BMC. External Pull-up resistor required.
SML1CLK / GPP_C6 / SUSWARN# / SUSPWRDNACK	I/OD	**System Management Link 1 Clock:** SMBus link to optional Embedded Controller or BMC. External Pull-up resistor required.
SML1DATA/ GPP_C7 / SUSACK#	I/OD	**System Management Link 1 Data:** SMBus link to optional Embedded Controller or BMC. External Pull-up resistor required.
SML1ALERT#/ PCHHOT# /GPP_B23 / GSPI1_CS1#	I/OD	**System Management 1 Alert:** Alert for the SMBus controller to optional Embedded Controller or BMC. A soft-strap determines the native function SML1ALERT# or PCHHOT# usage. This is **NOT** the right Alert pin for USB-C* usage. External Pull-up resistor is required on this pin.
PMCALERT#/ GPP_B11	I/OD	**USB Type-C* PD Controller / Re-timer Alert:** Alert for the SMLink1 Bus controller to all USB Type-C* PD Controllers, mandatory requirement for integrated USB-C* feature to work. External Pull-up resistor is required on this pin.
PCHHOT# / GPP_B23 / SML1ALERT# / GSPI1_CS1#	OD	This signal is used to indicate a PCH temperature out of bounds condition to an external EC. An external pull-up resistor is required on this signal.SML1ALERT# is IOD, PCHHOT# is OD

26.4 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
SML[1:0]ALERT#	Pull-down	20k ± 30%	The internal pull-down resistor is disable after RSMRST# de-asserted.
PCHHOT#	Pull-down	20k ± 30%	The internal pull-down resistor is disable after RSMRST# de-asserted.

26.5 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S3/S4/S5	Deep Sx
SML[1:0]DATA	Primary	Undriven	Undriven	Undriven	Undriven
SML[1:0]CLK	Primary	Undriven	Undriven	Undriven	Undriven
SML[1:0]ALERT#	Primary	Pull-down (Internal)	Driven Low	Pull-down (Internal)	OFF
PCHHOT#	Primary	Pull-down (Internal)	Driven Low	Pull-down (Internal)	OFF
PMCALERT#	Primary	Undriven	Undriven	Undriven	OFF

Notes:
1. Reset reference for primary well pins is RSMRST#.

26.6 Functional Description

The SMLink interfaces are controlled by the Intel® CSME.

SMLink0 is mainly used for integrated LAN. When an Intel LAN PHY is connected to SMLink0, a soft strap must be set to indicate that the PHY is connected to SMLink0. The interface will be running at the frequency of up to 1 MHz depending on different factors such as board routing or bus loading when the Fast Mode is enabled using a soft strap.

SMLink1 can be used with an Embedded Controller (EC) or Baseboard Management Controller (BMC).

Both SMLink0 and SMLink1 support up to 1 MHz.

26.6.1 Integrated USB-C Usage

SMLink1 is used to communicate with USB-C PD Controller on the platform to configure different modes such as USB, DP, Thunderbolt etc. When used for Integrated USB-C purposes, a soft strap must be set to indicate that integrated USB-C ports from CPU are being used.

SMLINK1 uses master mode and gets an alert signal from PMCALERT#.

Based on capabilities of different PD Controllers, re-timers needed for USB-C connector on the platform may need to be controlled by SoC also. In these cases, both PD Controller and Re-timers will be connected to SMLink1. SMLink1 is used for all USB-C connectors on the platform.

U-SKU supports 4 integrated USB-C ports and Y-SKU supports 3 integrated USB-C ports. Due to this, there could be maximum of 4 PD Controller and 4 re-timers. This translates to maximum of 8 devices on the SMLINK1 bus for a platform.

USB-C connectors are present on edges of systems and could also be on opposite ends, so (SMLink1, PMCAlert) could be routed to long distance on the motherboard provided total bus capacitance specification is met.

USB-C Re-timer control (like Firmware Load, USB-C configuration) handling depends on the number of I^2C ports available on the PD controller.

If the PD controller has 2 I^2C ports then PCH PMC will handle the Re-timer and PD controller, but if the PD controller has 3 or more I^2C ports then PCH PMC will handle only PD controller. Re-timers can be handled by PD controller.

SMLink1 should be run at 400KHz when used for USB-C purposes.

§ §

27 Host System Management Bus (SMBus) Controller

27.1 Acronyms

Acronyms	Description
ARP	Address Resolution Protocol
CRC	Cyclic Redundancy Check
PEC	Package Error Checking
SMBus	System Management Bus

27.2 References

Specification	Location
System Management Bus (SMBus) Specification, Version 2.0	http://www.smbus.org/specs/

27.3 Overview

The PCH provides a System Management Bus (SMBus) 2.0 host controller as well as an SMBus Slave Interface. The PCH is also capable of operating in a mode in which it can communicate with I^2C compatible devices.

The host SMBus controller supports up to 100 kHz clock speed.

27.4 Signal Description

Name	Type	Description
SMBCLK/ GPP_C0	I/OD	**SMBus Clock.** External Pull-up resistor is required.
SMBDATA/ GPP_C1	I/OD	**SMBus Data.** External Pull-up resistor is required.
SMBALERT#/ GPP_C2	I/OD	**SMBus Alert:** This signal is used to wake the system or generate SMI#. External Pull-up resistor is required.

27.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
SMBALERT#	Pull-down	20k ± 30%	The internal pull-down resistor is disable after RSMRST# de-asserted.

27.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S3/S4/S5	Deep Sx
SMBDATA	Primary	Undriven	Undriven	Undriven	Undriven
SMBCLK	Primary	Undriven	Undriven	Undriven	Undriven
SMBALERT#	Primary	Undriven	Undriven	Undriven	OFF

Notes:
1. Reset reference for primary well pins is RSMRST#.

27.7 Functional Description

The PCH provides an System Management Bus (SMBus) 2.0 host controller as well as an SMBus Slave Interface.

- **Host Controller:** Provides a mechanism for the processor to initiate communications with SMBus peripherals (slaves). The PCH is also capable of operating in a mode in which it can communicate with I^2C compatible devices.
- **Slave Interface:** Allows an external master to read from or write to the PCH. Write cycles can be used to cause certain events or pass messages, and the read cycles can be used to determine the state of various status bits. The PCH's internal host controller cannot access the PCH's internal Slave Interface.

27.7.1 Host Controller

The host SMBus controller supports up to 100-KHz clock speed and is clocked by the RTC clock.

The PCH can perform SMBus messages with either Packet Error Checking (PEC) enabled or disabled. The actual PEC calculation and checking is performed in SW. The SMBus host controller logic can automatically append the CRC byte if configured to do so.

The SMBus Address Resolution Protocol (ARP) is supported by using the existing host controller commands through software, except for the Host Notify command (which is actually a received message).

The PCH SMBus host controller checks for parity errors as a target. If an error is detected, the detected parity error bit in the PCI Status Register is set.

27.7.1.1 Host Controller Operation Overview

The SMBus host controller is used to send commands to other SMBus slave devices. Software sets up the host controller with an address, command, and, for writes, data and optional 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 SMI# or interrupt, if enabled.

The host controller supports 8 command protocols of the SMBus interface (Refer *System Management Bus (SMBus) Specification,* Version 2.0): Quick Command, Send Byte, Receive Byte, Write Byte/Word, Read Byte/Word, Process Call, Block Read/Write, and Block Write–Block Read Process Call.

The SMBus host controller requires that the various data and command fields be setup for the type of command to be sent. When 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, Host Command, Transmit Slave Address, Data 0, Data 1) should not be changed or read until the interrupt status message (INTR) has been set (indicating the completion of the command). Any register values needed for computation purposes should be saved prior to issuing of a new command, as the SMBus host controller updates all registers while completing the new command.

Slave functionality, including the Host Notify protocol, is available on the SMBus pins.

Using the SMB host controller to send commands to the PCH SMB slave port is not supported.

27.7.1.2 Command Protocols

In all of the following commands, the Host Status Register (offset 00h) is used to determine the progress of the command. While the command is in operation, the HOST_BUSY bit is set. If the command completes successfully, the INTR bit will be set in the Host Status Register. If the device does not respond with an acknowledge, and the transaction times out, the DEV_ERR bit is set.

If software sets the KILL bit in the Host Control Register while the command is running, the transaction will stop and the FAILED bit will be set after the PCH forces a time-out. In addition, if KILL bit is set during the CRC cycle, both the CRCE and DEV_ERR bits will also be set.

Quick Command

When programmed for a Quick Command, the Transmit Slave Address Register is sent. The PEC byte is never appended to the Quick Protocol. Software should force the PEC_EN bit to 0 when performing the Quick Command. Software must force the I2C_EN bit to 0 when running this command. Refer section 5.5.1 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

Send Byte/Receive Byte

For the Send Byte command, the Transmit Slave Address and Device Command Registers are sent. For the Receive Byte command, the Transmit Slave Address Register is sent. The data received is stored in the DATA0 register. Software must force the I2C_EN bit to 0 when running this command.

The Receive Byte is similar to a Send Byte, the only difference is the direction of data transfer. Refer sections 5.5.2 and 5.5.3 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

Write Byte/Word

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, Device Command, and Data0 Registers are sent. In addition, the Data1 Register is sent on a Write Word command. Software must force the I2C_EN bit to 0 when running this command. Refer section 5.5.4 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

Read Byte/Word

Reading data is slightly more complicated than writing data. First the PCH 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's address. The slave then returns 1 or 2 bytes of data. Software must force the I2C_EN bit to 0 when running this command.

When programmed for the read byte/word command, the Transmit Slave Address and Device Command Registers are sent. Data is received into the DATA0 on the read byte, and the DAT0 and DATA1 registers on the read word. Refer section 5.5.5 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

Process Call

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 simply a Write Word followed by a Read Word, but without a second command or stop condition.

When programmed for the Process Call command, the PCH transmits the Transmit Slave Address, Host Command, DATA0 and DATA1 registers. Data received from the device is stored in the DATA0 and DATA1 registers.

The Process Call command with I2C_EN set and the PEC_EN bit set produces undefined results. Software must force either I2C_EN or PEC_EN to 0 when running this command. Refer section 5.5.6 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

Note: For process call command, the value written into bit 0 of the Transmit Slave Address Register needs to be 0.

Note: If the I2C_EN bit is set, the protocol sequence changes slightly, the Command Code (Bits 18:11 in the bit sequence) are not sent. As a result, the slave will not acknowledge (Bit 19 in the sequence).

Block Read/Write

The PCH contains a 32-byte buffer for read and write data which can be enabled by setting bit 1 of the Auxiliary Control register at offset 0Dh in I/O space, 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 PCH, the interrupt is generated only after a transmission or reception of 32 bytes, or when the entire byte count has been transmitted/received.

The byte count field is transmitted but ignored by the PCH as software will end the transfer after all bytes it cares about have been sent or received.

For a Block Write, software must either force the I2C_EN bit or both the PEC_EN and AAC bits to 0 when running this command.

The block write begins with a slave address and a write condition. After the command code the PCH issues a byte count describing how many more bytes will follow in the message. If a slave had 20 bytes to send, the first byte would be the number 20 (14h), followed by 20 bytes of data. The byte count may not be 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, Device Command, and Data0 (count) registers are sent. Data is then sent from the Block Data Byte register; the total data sent being the value stored in the Data0 Register.

On block read commands, the first byte received is stored in the Data0 register, and the remaining bytes are stored in the Block Data Byte register. Refer section 5.5.7 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

Note: For Block Write, if the I2C_EN bit is set, the format of the command changes slightly. The PCH will still send the number of bytes (on writes) or receive the number of bytes (on reads) indicated in the DATA0 register. However, it will not send the contents of the DATA0 register as part of the message.When operating in I^2C mode (I2C_EN bit is set), the PCH will never use the 32-byte buffer for any block commands.

I^2C* Read

This command allows the PCH to perform block reads to certain I^2C devices, such as serial E^2PROMs. The SMBus Block Read supports the 7-bit addressing mode only.

However, this does not allow access to devices using the I^2C "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 I2C_EN bit. The I^2C Read command with the PEC_EN bit set produces undefined results. Software must force both the PEC_EN and AAC bit to 0 when running this command.

For I^2C Read command, the value written into bit 0 of the Transmit Slave Address Register (SMB I/O register, offset 04h) needs to be 0.

The format that is used for the command is shown in Table 27-1.

Table 27-1. I^2C* Block Read (Sheet 1 of 2)

Bit	Description
1	Start
8:2	Slave Address – 7 bits
9	Write
10	Acknowledge from slave
18:11	Send DATA1 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

Table 27-1. I²C* Block Read (Sheet 2 of 2)

Bit	Description
–	NOT Acknowledge
–	Stop

The PCH will continue reading data from the peripheral until the NAK is received.

Block Write–Block Read Process Call

The block write-block read process call 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 will be written in the first part of the message. If a master has 6 bytes to send, the byte count field will have 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 may differ 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 PEC byte. The PEC is computed on the total message beginning with the first slave address and using the normal PEC computational rules. It is highly recommended that a PEC byte be used with the Block Write-Block Read Process Call. Software must do a read to the command register (offset 2h) to reset the 32 byte buffer pointer prior to reading the block data register.

Note: There is no STOP condition before the repeated START condition, and that a NACK signifies the end of the read transfer.

Note: E32B bit in the Auxiliary Control register must be set when using this protocol.

Refer section 5.5.8 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

27.7.1.3 Bus Arbitration

Several masters may attempt to get on the bus at the same time by driving the SMBDATA line low to signal a start condition. The PCH continuously monitors the SMBDATA line. When the PCH is attempting to drive the bus to a 1 by letting go of the SMBDATA line, and it samples SMBDATA low, then some other master is driving the bus and the PCH will stop transferring data.

If the PCH sees that it has lost arbitration, the condition is called a collision. The PCH will set the BUS_ERR bit in the Host Status Register, and if enabled, generate an interrupt or SMI#. The processor is responsible for restarting the transaction.

27.7.1.4 Clock Stretching

Some devices may not be able to handle their clock toggling at the rate that the PCH as an SMBus master would like. They have the capability of stretching the low time of the clock. When the PCH attempts to release the clock (allowing the clock to go high), the clock will remain low for an extended period of time.

The PCH 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 can be stretched by an SMBus master if it is not ready to send or receive data.

27.7.1.5 Bus Timeout (PCH as SMBus Master)

If there is an error in the transaction, such that an SMBus device does not signal an acknowledge or holds the clock lower than the allowed Timeout time, the transaction will time out. The PCH will discard the cycle and set the DEV_ERR bit. The timeout minimum is 25 ms (800 RTC clocks). The Timeout counter inside the PCH will start after the first bit of data is transferred by the PCH and it is waiting for a response.

The 25-ms Timeout counter will not count under the following conditions:

1. BYTE_DONE_STATUS bit (SMBus I/O Offset 00h, Bit 7) is set
2. The SECOND_TO_STS bit (TCO I/O Offset 06h, Bit 1) is not set (this indicates that the system has not locked up).

27.7.1.6 Interrupts/SMI#

The PCH SMBus controller uses PIRQB# as its interrupt pin. However, the system can alternatively be set up to generate SMI# instead of an interrupt, by setting the SMBUS_SMI_EN bit.

Table 27-2, Table 27-3 and Table 27-4 specify how the various enable bits in the SMBus function control the generation of the interrupt, Host and Slave 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 will occur.

Table 27-2. Enable for SMBALERT#

Event	INTREN (Host Control I/O Register, Offset 02h, Bit 0)	SMB_SMI_EN (Host Configuration Register, D31:F4:Offset 40h, Bit 1)	SMBALERT_DIS (Slave Command I/O Register, Offset 11h, Bit 2)	Result
SMBALERT# asserted low (always reported in Host Status Register, Bit 5)	X	X	X	Wake generated
	X	1	0	Slave SMI# generated (SMBUS_SMI_STS)
	1	0	0	Interrupt generated

Table 27-3. Enables for SMBus Slave Write and SMBus Host Events

Event	INTREN (Host Control I/O Register, Offset 02h, Bit 0)	SMB_SMI_EN (Host Configuration Register, D31:F4:Offset 40h, Bit 1)	Event
Slave Write to Wake/SMI# Command	X	X	Wake generated when asleep. Slave SMI# generated when awake (SMBUS_SMI_STS).
Slave Write to SMLINK_SLAVE_SMI Command	X	X	Slave SMI# generated when in the S0 state (SMBUS_SMI_STS)
Any combination of Host Status Register [4:1] asserted	0	X	None
	1	0	Interrupt generated
	1	1	Host SMI# generated

Table 27-4. Enables for the Host Notify Command

HOST_NOTIFY_INTREN (Slave Control I/O Register, Offset 11h, Bit 0)	SMB_SMI_EN (Host Config Register, D31:F4:Off40h, Bit 1)	HOST_NOTIFY_WKEN (Slave Control I/O Register, Offset 11h, Bit 1)	Result
0	X	0	None
X	X	1	Wake generated
1	0	X	Interrupt generated
1	1	X	Slave SMI# generated (SMBUS_SMI_STS)

27.7.1.7 SMBus CRC Generation and Checking

If the AAC bit is set in the Auxiliary Control register, the PCH automatically calculates and drives CRC at the end of the transmitted packet for write cycles, and will check the CRC for read cycles. It will not transmit the contents of the PEC register for CRC. The PEC bit must not be set in the Host Control register if this bit is set, or unspecified behavior will result.

If the read cycle results in a CRC error, the DEV_ERR bit and the CRCE bit in the Auxiliary Status register at Offset 0Ch will be set.

27.7.2 SMBus Slave Interface

The PCH SMBus Slave interface is accessed using the SMBus. The SMBus slave logic will not generate or handle receiving the PEC byte and will only act as a Legacy Alerting Protocol device. The slave interface allows the PCH to decode cycles, and allows an external microcontroller to perform specific actions.

Key features and capabilities include:

- Supports decode of three types of messages: Byte Write, Byte Read, and Host Notify.
- Receive Slave Address register: This is the address that the PCH decodes. A default value is provided so that the slave interface can be used without the processor having to program this register.
- Receive Slave Data register in the SMBus I/O space that includes the data written by the external microcontroller.

- Registers that the external microcontroller can read to get the state of the PCH.
  - — Status bits to indicate that the SMBus slave logic caused an interrupt or SMI# Bit 0 of the Slave Status Register for the Host Notify command
  - — Bit 16 of the SMI Status Register for all others

Note: The external microcontroller should not attempt to access the PCH SMBus slave logic until either:

- — 800 milliseconds after both: RTCRST# is high and RSMRST# is high, OR
- — The PLTRST# de-asserts

If a master leaves the clock and data bits of the SMBus interface at 1 for 50 µs or more in the middle of a cycle, the PCH slave logic's behavior is undefined. This is interpreted as an unexpected idle and should be avoided when performing management activities to the slave logic.

27.7.2.1 Format of Slave Write Cycle

The external master performs Byte Write commands to the PCH SMBus Slave I/F. The "Command" field (bits 11:18) indicate which register is being accessed. The Data field (bits 20:27) indicate the value that should be written to that register.

Table 27-5 has the values associated with the registers.

Table 27-5. Slave Write Registers

Register	Function
0	Command Register. Refer Table 27-6 for valid values written to this register.
1–3	Reserved
4	Data Message Byte 0
5	Data Message Byte 1
6–7	Reserved
8	Reserved
9–FFh	Reserved
Note: The external microcontroller is responsible to make sure that it does not update the contents of the data byte registers until they have been read by the system processor. The PCH overwrites the old value with any new value received. A race condition is possible where the new value is being written to the register just at the time it is being read. The PCH will not attempt to cover this race condition (that is, unpredictable results in this case).	

Table 27-6. Command Types (Sheet 1 of 2)

Command Type	Description
0	Reserved
1	**WAKE/SMI#.** This command wakes the system if it is not already awake. If system is already awake, an SMI# is generated.
2	**Unconditional Powerdown.** This command sets the PWRBTNOR_STS bit, and has the same effect as the Powerbutton Override occurring.
3	**HARD RESET WITHOUT CYCLING:** This command causes a soft reset of the system (does not include cycling of the power supply). This is equivalent to a write to the CF9h register with Bits 2:1 set to 1, but Bit 3 set to 0.
4	**HARD RESET SYSTEM.** This command causes a hard reset of the system (including cycling of the power supply). This is equivalent to a write to the CF9h register with Bits 3:1 set to 1.

Table 27-6. Command Types (Sheet 2 of 2)

Command Type	Description
5	**Disable the TCO Messages.** This command will disable the PCH from sending Heartbeat and Event messages. Once this command has been executed, Heartbeat and Event message reporting can only be re-enabled by assertion and then de-assertion of the RSMRST# signal.
6	**WD RELOAD:** Reload watchdog timer.
7	Reserved
8	**SMLINK_SLV_SMI.** When the PCH detects this command type while in the S0 state, it sets the SMLINK_SLV_SMI_STS bit. This command should only be used if the system is in an S0 state. If the message is received during S3–S5 states, the PCH acknowledges it, but the SMLINK_SLV_SMI_STS bit does not get set. ***Note:*** It is possible that the system transitions out of the S0 state at the same time that the SMLINK_SLV_SMI command is received. In this case, the SMLINK_SLV_SMI_STS bit may get set but not serviced before the system goes to sleep. Once the system returns to S0, the SMI associated with this bit would then be generated. Software must be able to handle this scenario.
9–FFh	Reserved.

27.7.2.2 Format of Read Command

The external master performs Byte Read commands to the PCH SMBus Slave interface. The "Command" field (bits 18:11) indicate which register is being accessed. The Data field (bits 30:37) contain the value that should be read from that register.

Table 27-7. Slave Read Cycle Format

Bit	Description	Driven By	Comment
1	Start	External Microcontroller	
2–8	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
9	Write	External Microcontroller	Always 0
10	ACK	PCH	
11–18	Command code – 8 bits	External Microcontroller	Indicates which register is being accessed. Refer Table 27-8 for a list of implemented registers.
19	ACK	PCH	
20	Repeated Start	External Microcontroller	
21–27	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
28	Read	External Microcontroller	Always 1
29	ACK	PCH	
30–37	Data Byte	PCH	Value depends on register being accessed. Refer Table 27-8 for a list of implemented registers.
38	NOT ACK	External Microcontroller	
39	Stop	External Microcontroller	

Table 27-8. Data Values for Slave Read Registers (Sheet 1 of 2)

Register	Bits	Description
0	7:0	Reserved
1	2:0	**System Power State** 000 = S0 011 = S3 100 = S4 101 = S5 Others = Reserved
	7:3	Reserved
2	3:0	Reserved
	7:4	Reserved
3	5:0	**Watchdog Timer current value** ***Note:*** The Watchdog Timer has 10 bits, but this field is only 6 bits. If the current value is greater than 3Fh, the PCH will always report 3Fh in this field.
	7:6	Reserved
4	0	**Intruder Detect**.
	1	Reserved
	2	Reserved
	3	1 = **SECOND_TO_STS** bit set. This bit will be set after the second Timeout (SECOND_TO_STS bit) of the Watchdog Timer occurs.
	6:4	Reserved. Will always be 0, but software should ignore.
	7	**SMBALERT# Status.** Reflects the value of the SMBALERT# pin (when the pin is configured to SMBALERT#). Valid only if SMBALERT_DISABLE = 0. Value always returns 1 if SMBALERT_DISABLE = 1.
5	0	**FWH bad bit.** This bit will be 1 to indicate that the FWH read returned FFh, which indicates that it is probably blank.
	1	**Battery Low Status.** 1 if the BATLOW# pin a low.
	2	**SYS_PWROK Failure Status:** This bit will be 1 if the SYSPWR_FLR bit in the GEN_PMCON_2 register is set.
	3	Reserved
	4	Reserved
	5	**POWER_OK_BAD:** Indicates the failure core power well ramp during boot/resume. This bit will be active if the SLP_S3# pin is de-asserted and PCH_PWROK pin is not asserted.
	6	**Thermal Trip**: This bit will shadow the state of processor Thermal Trip status bit (CTS). Events on signal will not create a event message
	7	Reserved: Default value is "X" ***Note:*** Software should not expect a consistent value when this bit is read through SMBUS/SMLink
6	7:0	Contents of the Message 1 register.
7	7:0	Contents of the Message 2 register.
8	7:0	Contents of the WDSTATUS register.
9	7:0	Seconds of the RTC
A	7:0	Minutes of the RTC
B	7:0	Hours of the RTC
C	7:0	"Day of Week" of the RTC
D	7:0	"Day of Month" of the RTC
E	7:0	Month of the RTC

Table 27-8. Data Values for Slave Read Registers (Sheet 2 of 2)

Register	Bits	Description
F	7:0	Year of the RTC
10h–FFh	7:0	Reserved

27.7.2.2.1 Behavioral Notes

According to SMBus protocol, Read and Write messages always begin with a Start bit—Address—Write bit sequence. When the PCH detects that the address matches the value in the Receive Slave Address register, it will assume that the protocol is always followed and ignore the Write bit (Bit 9) and signal an Acknowledge during bit 10. In other words, if a Start—Address—Read occurs (which is invalid for SMBus Read or Write protocol), and the address matches the PCH's Slave Address, the PCH will still grab the cycle.

Also according to SMBus protocol, a Read cycle contains a Repeated Start—Address—Read sequence beginning at Bit 20. Once again, if the Address matches the PCH's Receive Slave Address, it will assume that the protocol is followed, ignore bit 28, and proceed with the Slave Read cycle.

27.7.2.3 Slave Read of RTC Time Bytes

The PCH SMBus slave interface allows external SMBus master to read the internal RTC's time byte registers.

The RTC time bytes are internally latched by the PCH's hardware whenever RTC time is not changing and SMBus is idle. This ensures that the time byte delivered to the slave read is always valid and it does not change when the read is still in progress on the bus. The RTC time will change whenever hardware update is in progress, or there is a software write to the RTC time bytes.

The PCH SMBus slave interface only supports Byte Read operation. The external SMBus master will read the RTC time bytes one after another. It is software's responsibility to check and manage the possible time rollover when subsequent time bytes are read.

For example, assuming the RTC time is 11 hours: 59 minutes: 59 seconds. When the external SMBus master reads the hour as 11, then proceeds to read the minute, it is possible that the rollover happens between the reads and the minute is read as 0. This results in 11 hours: 0 minute instead of the correct time of 12 hours: 0 minutes. Unless it is certain that rollover will not occur, software is required to detect the possible time rollover by reading multiple times such that the read time bytes can be adjusted accordingly if needed.

27.7.2.4 Format of Host Notify Command

The PCH 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 PCH already has data for a previously-received host notify command which has not been serviced yet by the host software (as indicated by the HOST_NOTIFY_STS bit), then it will 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 host software completely services the interrupt.

Note: Host software must always clear the HOST_NOTIFY_STS bit after completing any necessary reads of the address and data registers.

Table 27-9 shows the Host Notify format.

Table 27-9. Host Notify 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)	PCH	PCH NACKs if HOST_NOTIFY_STS is 1
17:11	Device Address – 7 bits	External Master	Indicates the address of the master; loaded into the Notify Device Address Register
18	Unused – Always 0	External Master	7-bit-only address; this bit is inserted to complete the byte
19	ACK	PCH	
27:20	Data Byte Low – 8 bits	External Master	Loaded into the Notify Data Low Byte Register
28	ACK	PCH	
36:29	Data Byte High – 8 bits	External Master	Loaded into the Notify Data High Byte Register
37	ACK	PCH	
38	Stop	External Master	

27.7.2.5 Format of Read Command

The external master performs Byte Read commands to the PCH SMBus Slave interface. The "Command" field (bits 18:11) indicate which register is being accessed. The Data field (bits 30:37) contain the value that should be read from that register.

Table 27-10. Slave Read Cycle Format (Sheet 1 of 2)

Bit	Description	Driven By	Comment
1	Start	External Microcontroller	
2–8	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
9	Write	External Microcontroller	Always 0
10	ACK	PCH	
11–18	Command code – 8 bits	External Microcontroller	Indicates which register is being accessed. Refer Table 27-11 for a list of implemented registers.
19	ACK	PCH	
20	Repeated Start	External Microcontroller	
21–27	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
28	Read	External Microcontroller	Always 1
29	ACK	PCH	
30–37	Data Byte	PCH	Value depends on register being accessed. Refer Table 27-11 for a list of implemented registers.

Table 27-10. Slave Read Cycle Format (Sheet 2 of 2)

Bit	Description	Driven By	Comment
38	NOT ACK	External Microcontroller	
39	Stop	External Microcontroller	

Table 27-11. Data Values for Slave Read Registers (Sheet 1 of 2)

Register	Bits	Description
0	7:0	Reserved for capabilities indication. Should always return 00h. Future chips may return another value to indicate different capabilities.
1	2:0	**System Power State** 000 = S0 011 = S3 100 = S4 101 = S5 Others = Reserved
	7:3	Reserved
2	3:0	Reserved
	7:4	Reserved
3	5:0	**Watchdog Timer current value** ***Note:*** The Watchdog Timer has 10 bits, but this field is only 6 bits. If the current value is greater than 3Fh, the PCH will always report 3Fh in this field.
	7:6	Reserved
4	0	**Intruder Detect.** Reserved
	1	**Temperature Event.** 1 = Temperature Event occurred. This bit will be set if the PCH's THRM# input signal is active. Else this bit will read "0."
	2	**DOA Processor Status**. This bit will be 1 to indicate that the processor is dead
	3	1 = **SECOND_TO_STS** bit set. This bit will be set after the second Timeout (SECOND_TO_STS bit) of the Watchdog Timer occurs.
	6:4	Reserved. Will always be 0, but software should ignore.
	7	**SMBALERT# Status.** Reflects the value of the GPIO11/SMBALERT# pin (when the pin is configured as SMBALERT#). Valid only if SMBALERT_DISABLE = 0. Value always return 1 if SMBALERT_DISABLE = 1. (high = 1, low = 0).
5	0	**FWH bad bit.** This bit will be 1 to indicate that the FWH read returned FFh, which indicates that it is probably blank.
	1	**Battery Low Status**. 1 if the BATLOW# pin is a 0.
	2	**SYS_PWROK Failure Status:** This bit will be 1 if the SYSPWR_FLR bit in the GEN_PMCON_2 register is set.
	3	Reserved
	4	Reserved
	5	**POWER_OK_BAD.** Indicates the failure core power well ramp during boot/resume. This bit will be active if the SLP_S3# pin is de-asserted and PCH_PWROK pin is not asserted.
	6	**Thermal Trip**. This bit will shadow the state of processor Thermal Trip status bit (CTS). Events on signal will not create a event message
	7	Reserved: Default value is "X" ***Note:*** Software should not expect a consistent value when this bit is read through SMBUS/SMLink
6	7:0	Contents of the Message 1 register.
7	7:0	Contents of the Message 2 register.

Table 27-11. Data Values for Slave Read Registers (Sheet 2 of 2)

Register	Bits	Description
8	7:0	Contents of the WDSTATUS register.
9	7:0	Seconds of the RTC
A	7:0	Minutes of the RTC
B	7:0	Hours of the RTC
C	7:0	"Day of Week" of the RTC
D	7:0	"Day of Month" of the RTC
E	7:0	Month of the RTC
F	7:0	Year of the RTC
10h–FFh	7:0	Reserved

Table 27-12. Enables for SMBus Slave Write and SMBus Host Events

Event	INTREN (Host Control I/O Register, Offset 02h, Bit 0)	SMB_SMI_EN (Host Configuration Register, D31:F3:Offset 40h, Bit 1)	Event
Slave Write to Wake/SMI# Command	X	X	Wake generated when asleep. Slave SMI# generated when awake (SMBUS_SMI_STS)
Slave Write to SMLINK_SLAVE_SMI Command	X	X	Slave SMI# generated when in the S0 state (SMBUS_SMI_STS)
Any combination of Host Status Register [4:1] asserted	0	X	None
	1	0	Interrupt generated
	1	1	Host SMI# generated

27.8 SMBus Power Gating

SMBus shares the Power Gating Domain with Primary-to-Sideband Bridge (P2SB).

A single FET controls the single Power Gating Domain; but SMBus and P2SB each has its own dedicated Power Gating Control Block.

The FET is only turned off when all these interfaces are ready to PG entry or already in the PG state.

§ §

28 Serial Peripheral Interface (SPI)

28.1 Acronyms

Acronyms	Description
CLK	Clock
CS	Chip Select
FCBA	Flash Component Base Address
FIBA	Flash Initialization Base Address
FLA	Flash Linear Address
FMBA	Flash Master Base Address
FPSBA	Flash PCH Strap Base Address
FRBA	Flash Region Base Address
MDTBA	MIP Descriptor Table Base Address
MISO	Mater In Slave Out
MOSI	Mater Out Slave In
TPM	Trusted Platform Module

28.2 Feature Overview

The PCH provides two Serial Peripheral Interfaces (SPI). The SPI0 interface consists of 3 Chip Select signals. It is allowing up to two flash memory devices (SPI0_CS0# and SPI0_CS1#) and one TPM device (SPI0_CS2#) to be connected to the PCH. The SPI1 interface implementing 1 Chip Select signal (SPI1_CS#), is intended for integrated touch implementation. The SPI0 interface support either 1.8V or 3.3V. But SPI1 interface supports 1.8V only.The voltage is selected via a strap on INTRUDER# signal. Refer Section 28.7 for VCCSPI Voltage (3.3V or 1.8V) Selection.

28.3 Signal Description

Name	Type	Description
SPI0_CLK	O	**SPI0 Clock**: SPI clock signal for the common flash/TPM interface. Supports 20 MHz, 33 MHz and 50 MHz.
SPI0_CS0#	O	**SPI0 Chip Select 0**: Used to select the primary SPI0 Flash device. ***Note:*** This signal cannot be used for any other type of device than SPI Flash.
SPI0_CS1#	O	**SPI0 Chip Select 1**: Used to select an optional secondary SPI0 Flash device. ***Note:*** This signal cannot be used for any other type of device than SPI Flash.

Name	Type	Description
SPI0_CS2#	O	**SPI0 Chip Select 2:** Used to select the TPM device if it is connected to the SPI0 interface. It cannot be used for any other type of device.
SPI0_MOSI	I/O	**SPI0 Master OUT Slave IN**: Defaults as a data output pin for PCH in Dual Output Fast Read mode. Can be configured with a Soft Strap as a bidirectional signal (SPI0_IO0) to support the Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes.
SPI0_MISO	I/O	**SPI0 Master IN Slave OUT**: Defaults as a data input pin for PCH in Dual Output Fast Read mode. Can be configured with a Soft Strap as a bidirectional signal (SPI0_IO1) to support the Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes.
SPI0_IO2	I/O	**SPI0 Data I/O:** A bidirectional signal used to support Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes. This signal is not used in Dual Output Fast Read mode.
SPI0_IO3	I/O	**SPI0 Data I/O:** A bidirectional signal used to support Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes. This signal is not used in Dual Output Fast Read mode.
SPI1_CLK/GPP_E11 / BK1 / SBK1	O	**SPI1 Clock**: SPI1 Clock output from PCH
SPI1_MISO/GPP_E12 / BK2 / SBK2	I/O	**SPI1 Master IN Slave OUT**: SPI1 serial input data from the SPI1 Touch Screen device to PCH. This Pin will also function as Output during Dual and Quad I/O operation
SPI1_MOSI/GPP_E13 / BK3 / SBK3	I/O	**SPI1 Master OUT Slave IN**: SPI1 serial output data from PCH to the SPI1 Touch Screen device. This Pin will also function as Input during Dual and Quad I/O operation
SPI1_IO2/GPP_E1	I/O	**SPI1 Data I/O:** SPI1 I/O to comprehend the support for the Quad I/O operation
SPI1_IO3/GPP_E2	I/O	**SPI1 Data I/O:** SPI1 I/O to comprehend the support for the Quad I/O operation
SPI1_CS1#/GPP_E8 / SATALED#	O	**SPI1 Chip Select**: SPI1 chip select

28.4 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
SPI0_CLK	Pull-down	20k ± 30%	
SPI0_MOSI	Pull-up	20k ± 30%	Note
SPI0_MISO	Pull-up	20k ± 30%	Note
SPI0_CS[2:0]#	Pull-down	20k ± 30%	
SPI0_IO[2:3]	Pull-up	20k ± 30%	Note

Note: The internal pull-up is disabled when RSMRST# is asserted (during reset) and only enabled after RSMRST# de-assertion

28.5 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S3/S4/S5	Deep Sx
SPI0_CLK	Primary	Internal Pull-down	Driven Low	Driven Low	Off
SPI0_MOSI	Primary	Hi-Z (Refer Note 2)	Internal PU, then Driven Low	Driven Low	Off
SPI0_MISO	Primary	Hi-Z	Internal Pull-up	Internal Pull-up	Off
SPI0_CS0#	Primary	Internal Pull-down	Driven High	Driven High	Off
SPI0_CS1#	Primary	Internal Pull-down	Driven High	Driven High	Off
SPI0_CS2#	Primary	Internal Pull-down	Driven High	Driven High	Off
SPI0_IO[3:2]	Primary	Hi-Z (Refer Note 2)	Internal Pull-up	Internal Pull-up	Off

Notes:
1. Reset reference for primary well pins is RSMRST#.
2. SPI0_MOSI, SPI0_IO[3:2] also function as strap pins. The actual pin state during Reset is dependent on the platform Pull-up/Pull-down resistor.

28.6 Functional Description

28.6.1 SPI0 for Flash

28.6.1.1 Overview

The Serial Peripheral Interface (SPI0) supports 2 SPI flash devices via 2 chip select (SPI0_CS0# and SPI0_CS1#). The maximum size of flash supported is determined by the SFDP-discovered addressing capability of each device. Each component can be up to 16 MB (32 MB total addressable) using 3-byte addressing. Each component can be up to 64 MB (128 MB total addressable) using 4-byte addressing. Another chip select (SPI0_CS2#) is also available and only used for TPM on SPI support. PCH drives the SPI0 interface clock at either 20 MHz, 33 MHz, or 50 MHz and will function with SPI flash devices that support at least one of these frequencies.

A SPI0 flash device supporting SFDP (Serial Flash Discovery Parameter) is required for all PCH design. A SPI0 flash device on SPI0_CS0# with a valid descriptor MUST be attached directly to the PCH.

The PCH supports fast read which consist of:

1. Dual Output Fast Read (Single Input Dual Output)
2. Dual I/O Fast Read (Dual Input Dual Output)

3. Quad Output Fast Read (Single Input Quad Output)
4. Quad I/O Fast Read (Quad Input Quad Output)

The PCH SPI0 has a third chip select SPI0_CS2# for TPM support over SPI. The TPM on SPI0 will use SPI0_CLK, SPI0_MISO, SPI0_MOSI and SPI0_CS2# SPI signals.

28.6.1.2 SPI0 Supported Features

28.6.1.2.1 Descriptor Mode

Descriptor Mode is required for all SKUs of the PCH. Non-Descriptor Mode is not supported.

28.6.1.2.2 SPI0 Flash Regions

In Descriptor Mode the Flash is divided into five separate regions.

Table 28-1. SPI0 Flash Regions

Region	Content
0	Flash Descriptor
1	BIOS
2	Intel Management Engine
3	Gigabit Ethernet
4	Platform Data
5	EC

Only four masters can access the regions: Host processor running BIOS code, Integrated Gigabit Ethernet and Host processor running Gigabit Ethernet Software, Intel® Management Engine, and the EC.

The Flash Descriptor and Intel® CSME region are the only required regions. The Flash Descriptor has to be in region 0 and region 0 must be located in the first sector of Device 0 (Offset 0). All other regions can be organized in any order.

Regions can extend across multiple components, but must be contiguous.

Flash Region Sizes

SPI0 flash space requirements differ by platform and configuration. The Flash Descriptor requires one 4-KB or larger block. GbE requires two 4-KB or larger blocks. The amount of flash space consumed is dependent on the erase granularity of the flash part and the platform requirements for the Intel® CSME and BIOS regions. The Intel® CSME region contains firmware to support Intel Active Management Technology and other Intel® CSME capabilities.

Table 28-2. Region Size Versus Erase Granularity of Flash Components (Sheet 1 of 2)

Region	Size with 4-KB Blocks	Size with 8-KB Blocks	Size with 64-KB Blocks
Descriptor	4 KB	8 KB	64 KB
GbE	8 KB	16 KB	128 KB
BIOS	Varies by Platform	Varies by Platform	Varies by Platform

Table 28-2. Region Size Versus Erase Granularity of Flash Components (Sheet 2 of 2)

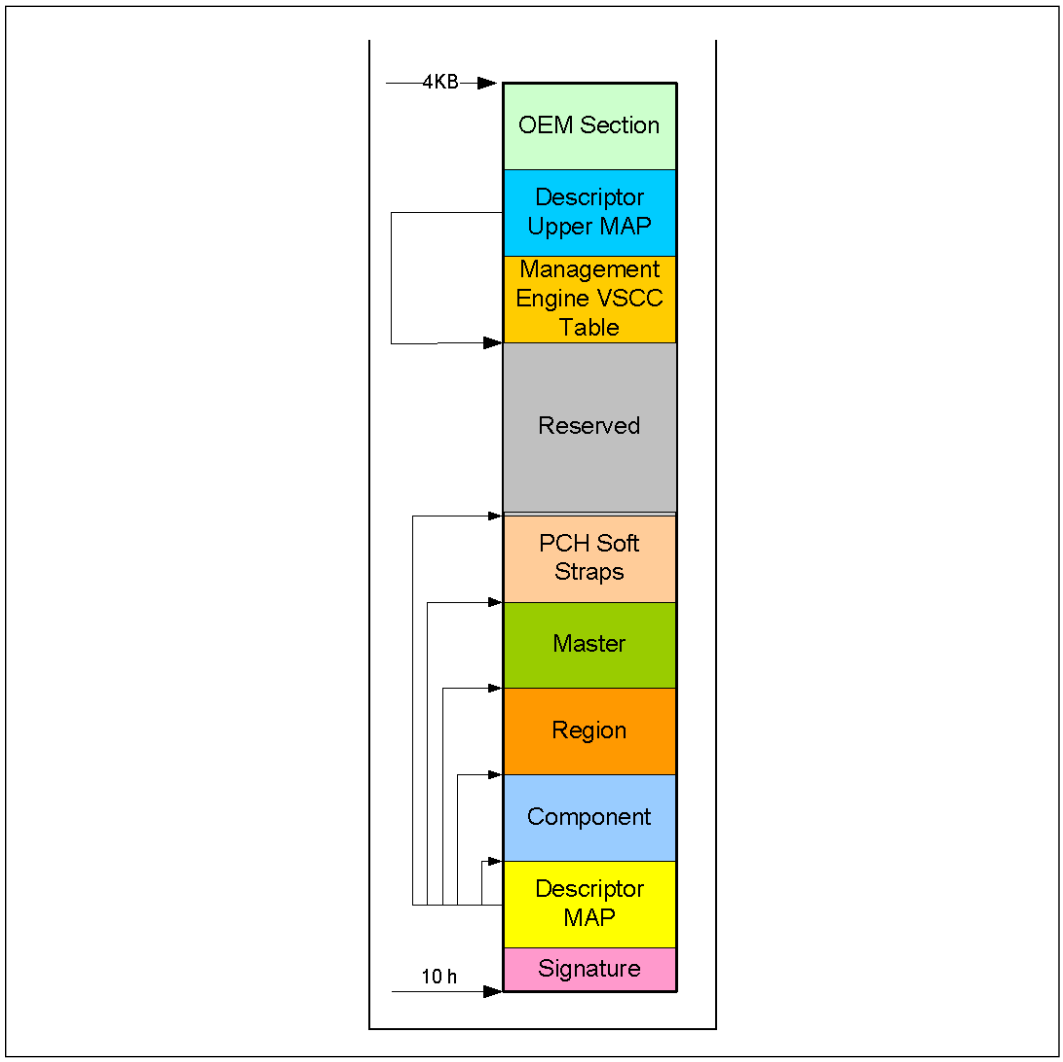
Region	Size with 4-KB Blocks	Size with 8-KB Blocks	Size with 64-KB Blocks
Intel® CSME	Varies by Platform	Varies by Platform	Varies by Platform
EC	Varies by Platform	Varies by Platform	Varies by Platform

28.6.1.3 Flash Descriptor

The bottom sector of the flash component 0 contains the Flash Descriptor. The maximum size of the Flash Descriptor is 4 KB. If the block/sector size of the SPI0 flash device is greater than 4 KB, the flash descriptor will only use 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 can only be written during the manufacturing process as its read/write permissions must be set to read only when the computer leaves the manufacturing floor.

The Flash Descriptor is made up of eleven sections as shown in Figure 28-1.

Figure 28-1. Flash Descriptor Regions



- The Flash signature selects Descriptor Mode as well as verifies if the flash is programmed and functioning. The data at the bottom of the flash (offset 10h) must be 0FF0A55Ah in order to be in Descriptor mode.
- The Descriptor map has pointers to the other five descriptor sections as well as the size of each.
- The component section has information about the SPI0 flash in the system including: the number of components, density of each, invalid instructions (such as chip erase), and frequencies for read, fast read and write/erase instructions.
- The Region section points to the three other regions as well as the size of each region.
- The master region contains the security settings for the flash, granting read/write permissions for each region and identifying each master by a requester ID.
- The processor and PCH Soft Strap sections contain processor and PCH configurable parameters.

- The Reserved region between the top of the processor strap section and the bottom of the OEM Section is reserved for future chipset usages.
- The Descriptor Upper MAP determines the length and base address of the Management Engine VSCC Table.
- The Management Engine VSCC Table holds the JEDEC ID and the VSCC information of the entire SPI0 Flash supported by the NVM image.
- OEM Section is 256 bytes reserved at the top of the Flash Descriptor for use by OEM.

28.6.1.3.1 Descriptor Master Region

The master region defines read and write access setting for each region of the SPI0 device. The master region recognizes four masters: BIOS, Gigabit Ethernet, Management Engine, and EC. Each master is only allowed to do direct reads of its primary regions.

Table 28-3. Region Access Control Table

Master Read/Write Access				
Region	**Processor and BIOS**	**Intel® CSME**	**GbE Controller**	**EC**
BIOS	Read/Write	N/A	Read/Write	Note
Intel® Management Engine (ME)	N/A	Read/Write	Read	N/A
Gigabit Ethernet	N/A	N/A	Read/Write	N/A
EC	Read	N/A	N/A	Read/Write
Note: Optional BIOS access to the EC region.				

28.6.1.3.2 Flash Descriptor CPU Complex Soft Strap Section

Region Name	Starting Address
Signature	10h
Component FCBA	30h
Regions FRBA	40h
Masters FMBA	80h
PCH Straps FPSBA	100h
MDTBA	C00h
PMC Straps	C14h
CPU Straps	C2Ch
Intel® CSME Straps	C3Ch
Register Init FIBA	340h

28.6.1.4 Flash Access

There are two types of accesses: Direct Access and Program Register Accesses.

28.6.1.4.1 Direct Access

- Masters are allowed to do direct read only of their primary region
  - — Gigabit Ethernet region can only be directly accessed by the Gigabit Ethernet controller. Gigabit Ethernet software must use Program Registers to access the Gigabit Ethernet region.
- Master's Host or Management Engine virtual read address is converted into the SPI0 Flash Linear Address (FLA) using the Flash Descriptor Region Base/Limit registers

Direct Access Security

- Requester ID of the device must match that of the primary Requester ID in the Master Section
- Calculated Flash Linear Address must fall between primary region base/limit
- Direct Write not allowed
- Direct Read Cache contents are reset to 0's on a read from a different master

28.6.1.4.2 Program Register Access

- Program Register Accesses are not allowed to cross a 4-KB boundary and can not issue a command that might extend across two components
- Software programs the FLA corresponding to the region desired
  - — Software must read the devices Primary Region Base/Limit address to create a FLA.

Register Access Security

- Only primary region masters can access the registers

28.6.2 SPI0 Support for TPM

The PCH's SPI0 flash controller supports a discrete TPM on the platform via its dedicated SPI0_CS2# signal. The platform must have no more than 1 TPM.

SPI0 controller supports accesses to SPI0 TPM at approximately 17 MHz, 33 MHz and 48 MHz depending on the PCH soft strap. 20 MHz is the reset default, a valid PCH soft strap setting overrides the requirement for the 20 MHz. SPI0 TPM device must support a clock of 20 MHz, and thus should handle 15-20 MHz. It may but is not required to support a frequency greater than 20 MHz.

TPM requires the support for the interrupt routing. However, the TPM's interrupt pin is routed to the PCH's PIRQ pin. Thus, TPM interrupt is completely independent from the SPI0 controller.

28.6.3 SPI1 Support for Touch Device

The SPI1 controller is dedicated for supporting SPI1 touch device via 1 chip select (SPI1_CS#) with Quad IO. The SPI1 interface supports only 1.8V. PCH drives the SPI1 interface clock at 30 MHz and will function with SPI Touch device that support at this frequency.

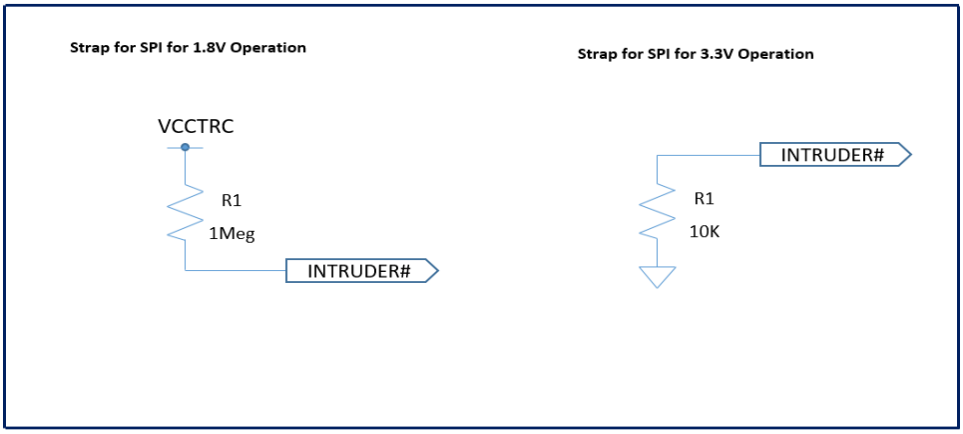
28.7 VCCSPI Voltage (3.3 V or 1.8 V) Selection

The VCCSPI voltage (3.3 V or 1.8 V) is selected via a hard strap on the INTRUDER#.This strap sets the SPI interface signaling voltage at the rising edge of RTCRST#. Designers should strap this pin to match the expected interface operational voltage for their target SPI device as follows:

0 = SPI interface operation voltage is 3.3 V (ground through a 10 kohm resistor)

1 = SPI interface operation voltage is 1.8 V (pulled up with 1 Mohm to VCCRTC)

Figure 28-2. VCCSPI Voltage (3.3 V or 1.8 V) Selection



§ §

29 Intel® Serial I/O Generic SPI (GSPI) Controllers

29.1 Acronyms

Acronyms	Description
GSPI	Generic Serial Peripheral Interface
LTR	Latency Tolerance Reporting

29.2 Feature Overview

The PCH implements three generic SPI interfaces to support devices that uses serial protocol for transferring data.

Each interface consists of a clock (CLK), two chip selects (CS) and 2 data lines (MOSI and MISO).

The GSPI interfaces support the following features:

- Support bit rates up to 20 Mbits/s
- Support data size from 4 to 32 bits in length and FIFO depths of 64 entries
- Support DMA with 128-byte FIFO per channel (up to 64-byte burst)
- Full duplex synchronous serial interface
- Support the Motorola's SPI protocol
- Operate in master mode only

Note: Slave mode is not supported.

29.3 Signal Description

Name	Type	Description
GSPI0_CS0#/ GPP_B15	O	**Generic SPI 0 Chip Select 0**
GSPI0_CS1#/ GPP_B14 / SPKR / TIME_SYNC1	O	**Generic SPI 0 Chip Select 1**
GSPI0_CLK/ GPP_B16	O	**Generic SPI 0 Clock**
GSPI0_MISO/ GPP_B17	I	**Generic SPI 0 MISO**
GSPI0_MOSI/ GPP_B18	O	**Generic SPI 0 MOSI** ***Note:*** This signal is also utilized as a strap. Refer the pin strap section for more info.
GSPI1_CS0#/ GPP_B19	O	**Generic SPI 1 Chip Select 0**

Name	Type	Description
GSPI1_CS1# / GPP_B23 / SML1ALERT# / PCHHOT#	O	**Generic SPI 1 Chip Select 1**
GSPI1_CLK/ GPP_B20	O	**Generic SPI 1 Clock**
GSPI1_MISO/ GPP_B21	I	**Generic SPI 1 MISO**
GSPI1_MOSI/ GPP_B22	O	**Generic SPI 1 MOSI** ***Note:*** This signal is also utilized as a strap. Refer the pin strap section for more info.
GSPI2_CS0#/ GPP_D9 / ISH_SPI_CS# / DDP3_CTRLCLK / TBT_LSX2_TXD	O	**Generic SPI 2 Chip Select 0**
GSPI2_CS1# / GPP_D15 / ISH_UART0_RTS# / IMGCLKOUT5	O	**Generic SPI 2 Chip Select 1**
GSPI2_CLK/ GPP_D10 / ISH_SPI_CLK / DDP3_CTRLDATA / TBT_LSX2_RXD	O	**Generic SPI 2 Clock**
GSPI2_MISO/ GPP_D11 / ISH_SPI_MISO / TBT_LSX3_TXD	I	**Generic SPI 2 MISO**
GSPI2_MOSI/ GPP_D12 / ISH_SPI_MOSI / TBT_LSX3_RXD	O	**Generic SPI 2 MOSI**

29.4 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
GSPI0_MOSI	Pull Down	20k ± 30%	The integrated pull down is disabled after PCH_PWROK assertion
GSPI1_MOSI	Pull Down	20k ± 30%	The integrated pull down is disabled after PCH_PWROK assertion
GSPI2_MOSI	Pull Down	20k ± 30%	The integrated pull down is disabled after PCH_PWROK assertion
GSPI0_MISO	Pull Down	20k ± 30%	
GSPI1_MISO	Pull Down	20k ± 30%	
GSPI2_MISO	Pull Down	20k ± 30%	

29.5 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S3/S4/S5	Deep Sx
GSPI0_CS0#, GSPI0_CS1#, GSPI1_CS0#, GSPI1_CS1#, GSPI2_CS0#, GSPI2_CS1#	Primary	Undriven	Undriven	Undriven	OFF
GSPI2_CLK, GSPI1_CLK, GSPI0_CLK	Primary	Undriven	Undriven	Undriven	OFF
GSPI2_MISO, GSPI1_MISO, GSPI0_MISO	Primary	Undriven	Undriven	Undriven	OFF
GSPI2_MOSI, GSPI1_MOSI, GSPI0_MOSI	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF

Notes:
3. Reset reference for primary well pins is RSMRST#.

29.6 Functional Description

29.6.1 Controller Overview

The generic SPI controllers can only be set to operate as a master.

The processor or DMA accesses data through the GSPI port's transmit and receive FIFOs.

A processor access takes the form of programmed I/O, transferring one FIFO entry per access. Processor accesses must always be 32 bits wide. Processor writes to the FIFOs are 32 bits wide, but the PCH will ignore all bits beyond the programmed FIFO data size. Processor reads to the FIFOs are also 32 bits wide, but the receive data written into the Receive FIFO is stored with '0' in the most significant bits (MSB) down to the programmed data size.

The FIFOs can also be accessed by DMA, which must be in multiples of 1, 2, or 4 bytes, depending upon the EDSS value, and must also transfer one FIFO entry per access.

For writes, the Enhanced SPI takes the data from the transmit FIFO, serializes it, and sends it over the serial wire to the external peripheral. Receive data from the external peripheral on the serial wire is converted to parallel words and stored in the receive FIFO.

A programmable FIFO trigger threshold, when exceeded, generates an interrupt or DMA service request that, if enabled, signals the processor or DMA respectively to empty the Receive FIFO or to refill the Transmit FIFO.

The GSPI controller, as a master, provides the clock signal and controls the chip select line. Commands codes as well as data values are serially transferred on the data signals. The PCH asserts a chip select line to select the corresponding peripheral device with which it wants to communicate. The clock line is brought to the device whether it is selected or not. The clock serves as synchronization of the data communication.

29.6.2 DMA Controller

The GSPI controllers have an integrated DMA controller.

29.6.2.1 DMA Transfer and Setup Modes

The DMA can operate in the following modes:

1. Memory to peripheral transfers. This mode requires that the peripheral control the flow of the data to itself.
2. Peripheral to memory transfer. This mode requires that the peripheral control the flow of the data from itself.

The DMA supports the following modes for programming:

1. Direct programming. Direct register writes to DMA registers to configure and initiate the transfer.
2. Descriptor based linked list. The descriptors will be stored in memory. The DMA will be informed with the location information of the descriptor. DMA initiates reads and programs its own register. The descriptors can form a linked list for multiple blocks to be programmed.
3. Scatter Gather mode.

29.6.2.2 Channel Control

- The source transfer width and destination transfer width are programmable. The width can be programmed to 1, 2, or 4 bytes.
- Burst size is configurable per channel for source and destination. The number is a power of 2 and can vary between 1,2,4,...,128. this number times the transaction width gives the number of bytes that will be transferred per burst.
- Individual Channel enables. If the channel is not being used, then it should be clock gated.
- Programmable Block size and Packing/Unpacking. Block size of the transfer is programmable in bytes. the block size is not limited by the source or destination transfer widths.
- Address incrementing modes: The DMA has a configurable mechanism for computing the source and destination addresses for the next transfer within the current block. The DMA supports incrementing addresses and constant addresses.
- Flexibility to configure any hardware handshake sideband interface to any of the DMA channels.
- Early termination of a transfer on a particular channel.

29.6.3 Reset

Each host controller has an independent rest associated with it. Control of these resets is accessed through the Reset Register.

Each host controller and DMA will be in reset state once powered ON and require SW (BIOS or driver) to write into the corresponding reset register to bring the controller from reset state into operational mode.

29.6.4 Power Management

29.6.4.1 Device Power Down Support

In order to power down peripherals connected to the PCH GSPI bus, the idle configured state of the I/O signals must be retained to avoid transitions on the bus that can affect the connected powered peripheral. Connected devices are allowed to remain in the D0 active or D2 low power states when the bus is powered off (power gated). The PCH HW will prevent any transitions on the serial bus signals during a power gate event.

29.6.4.2 Latency Tolerance Reporting (LTR)

Latency Tolerance Reporting is used to allow the system to optimize internal power states based on dynamic data, comprehending the current platform activity and service latency requirements. However, the GSPI bus architecture does not provide the architectural means to define dynamic latency tolerance messaging. Therefore, the interface supports this by reporting its service latency requirements to the platform power management controller via LTR registers.

The controller's latency tolerance reporting can be managed by one of the two following schemes. The platform integrator must choose the correct scheme for managing latency tolerance reporting based on the platform, OS and usage.

1. Platform/HW Default Control. This scheme is used for usage models in which the controller's state correctly informs the platform of the current latency requirements. In this scheme, the latency requirement is a function of the controller state. The latency for transmitting data to/from its connected device at a given rate while the controller is active is representative of the active latency requirements. On the other hand if the device is not transmitting or receiving data and idle, there is no expectation for end to end latency.
2. Driver Control. This scheme is used for usage models in which the controller state does not inform the platform correctly of the current latency requirements. If the FIFOs of the connected device are much smaller than the controller FIFOs, or the connected device's end-to-end traffic assumptions are much smaller than the latency to restore the platform from low power state, driver control should be used.

29.6.5 Interrupts

GSPI interface has an interrupt line which is used to notify the driver that service is required.

When an interrupt occurs, the device driver needs to read both the host controller and DMA interrupt status and transmit completion interrupt registers to identify the interrupt source. Clearing the interrupt is done with the corresponding interrupt register in the host controller or DMA.

All interrupts are active high and their behavior is level interrupt.

29.6.6 Error Handling

Errors that might occur on the external GSPI signals are comprehended by the host controller and reported to the interface host controller driver through the MMIO registers.

§ §

30 Testability

30.1 Acronyms

Acronyms	Description
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input/Output
I/OD	Input/Output Open Drain
JTAG	Joint Test Action Group
DCI	Direct Connect Interface
BSDL	Boundary Scan Description Language
DbC	Debug Class Devices

30.2 References

Specification	Location
IEEE Standard Test Access Port and Boundary Scan Architecture	http://standards.ieee.org/findstds/standard/1149.1-2013.html

30.3 Feature Overview

JTAG:

This section contains information regarding the testability signals that provides access to JTAG, run control, system control, and observation resources. JTAG (TAP) ports are compatible with the IEEE Standard Test Access Port and Boundary Scan Architecture 1149.1 and 1149.6 Specification, as detailed per device in each BSDL file. JTAG Pin definitions are from IEEE Standard Test Access Port and Boundary Scan. Architecture (IEEE Std. 1149.1-2001).

Intel® Trace Hub:

Intel® Trace Hub is a debug architecture that unifies hardware and software system visibility. Intel® Trace Hub is not merely intended for hardware debug or software debug, but full system debug. This includes debugging hardware and software as they interact and produce complex system behavior. Intel® Trace Hub defines new features and also leverages some existing debug technologies to provide a complete framework for hardware and software co-debug, software development and tuning, as well as overall system performance optimization.

Intel® Trace Hub is a set of silicon features with supported software API. The primary purpose is to collect trace data from different sources in the system and combine them into a single output stream with time-correlated to each other. Intel® Trace Hub uses common hardware interface for collecting time-correlated system traces through standard destinations. Intel® Trace Hub adopts industry standard (MIPI* STPv2) debug methodology for system debug and software development.

There are multiple destinations to receive the trace data from Intel® Trace Hub:

- Direct Connect Interface (DCI)
  - — OOB Hosting DCI
  - — USB 3.2 hosting DCI.DBC
- System Memory

There are multiple trace sources planned to be supported in the platform:

- BIOS
- Intel® CSME
- AET (Architecture Event Trace)
- Power Management Event Trace
- Windows* ETW (for driver or application)

30.4 JTAG

30.4.1 Signal Description

Table 30-1. Testability Signals

Name	Type	Description
PCH_TCK	I/O	**Test Clock Input (TCK):** The test clock input provides the clock for the JTAG test logic.
PCH_TMS	I/OD	**Test Mode Select (TMS):** The signal is decoded by the Test Access Port (TAP) controller to control test operations.
PCH_TDI	I/OD	**Test Data Input (TDI):** Serial test instructions and data are received by the test logic at TDI.
PCH_TDO	I/OD	**Test Data Output (TDO):** TDO is the serial output for test instructions and data from the test logic defined in this standard.
PCH_JTAGX	I/O	This pin is used to support merged debug port topologies.
PCH_TRST#	O	JTAG output from DCI to CPU
DBG_PMODE	O	Power Mode Indicator. This signal is used to transmit processor and PCH power/reset information to the Debugger.

30.4.2 I/O Signal Planes and States

Table 30-2. Power Planes and States for Testability Signals (Sheet 1 of 2)

Signal Name	Power Plane	Resistors	During Reset[2]	Immediately after Reset[2]	S3/S4/S5	Deep Sx
PCH_TCK	Primary	Strong Internal Pull-Down	L	L	L	OFF
PCH_TMS	Primary	Internal Pull-Up	H	H	H	OFF
PCH_TDI	Primary	Internal Pull-Up	H	H	H	OFF
PCH_TDO	Primary	External Pull-Up	Z	Z	Z	OFF
PCH_JTAGX[1]	Primary	Internal Strong Pull-Up (as TDO Input), Internal Strong Pull-Down (as TCK Output)	H	H/L	H/L	OFF

Table 30-2. Power Planes and States for Testability Signals (Sheet 2 of 2)

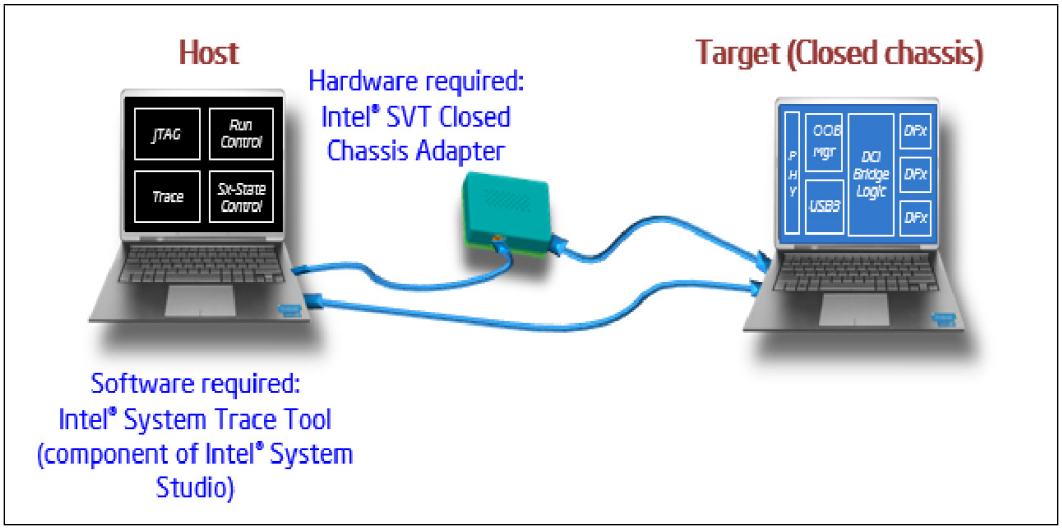
PCH_TRST#	Primary	Internal Strong Pull-Down	L	L	L	OFF
DBG_PMODE	Primary	Internal Pull-Up	H	H	H	OFF

Note:
1. This signal is used in common JTAG topology to take in last device's TDO to DCI. The only planned supported topology is the Shared Topology. Thus, this pin will operate as TCK mode.
2. Reset reference for primary well pins is RSMRST#.

30.5 Intel® Trace Hub (Intel® TH)

30.5.1 Platform Setup

Figure 30-1. Platform Setup with Intel® Trace Hub



30.6 Direct Connect Interface (DCI)

Direct Connect Interface (DCI) is a new debug transport technology to enable closed chassis debug through any of USB 3.2 ports out from Intel silicon. Some bridging logic is embedded in the silicon to "bridge" the gap between standard I/O ports and the debug interfaces including JTAG, probe mode, hooks, trace infrastructure, and etc. To control the operation of this embedded logic, a DCI packet based protocol is invented which controls and data can be sent or received. This protocol can operate over a few different physical transport paths to the target which known as "hosting interfaces".

Note: DCI and USB 3.2 based debugger (kernel level debugger) are mutually exclusive.

There are two types of DCI hosting interfaces in the platform:

- OOB Hosting DCI
- USB 3.2 Hosting DCI.DBC

Supported capabilities in DCI are:

- Closed Chassis Debug at S0 and Sx State
- JTAG Access and Run Control (Probe Mode)

- System Tracing with Intel® Trace Hub

Debug host software that support DCI are:

- Intel® ITP II Platform Debug Toolkit (PDT)
- Intel® System Studio (ISS)

30.6.1 Out Of Band (OOB) Hosting DCI

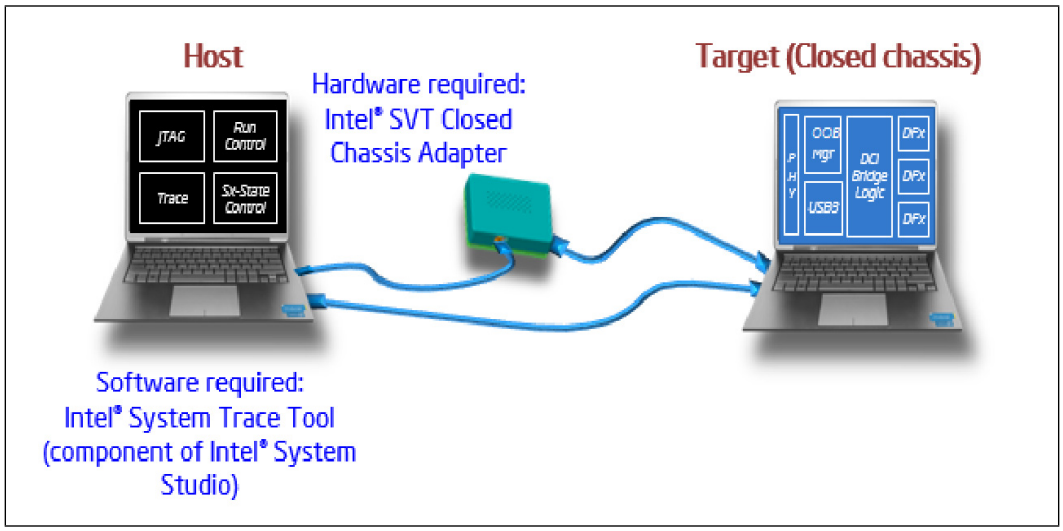
OOB was developed to provide an alternate path to convey controls and data to or from the EXI/DCI by connecting physically to the target through a USB 3.2 port. OOB provides an alternate side band path around the USB 3.2 controller, so that the embedded logic can be accessed, even when the USB 3.2 controller is not alive (such as in low power states) or is malfunctioning. This path does not rely on USB 3.2 protocol, link layer, or physical layer, because the xHCI functions are generally not available in such conditions. Instead, this path relies on a special adapter that was developed by Intel called the Intel® SVT Closed Chassis Adapter (CCA). It is a simple data transformation device. This adapter generates a OOB signaling protocol operating at up to 400 MHz and serializes data flowing through it. This adapter works together with debug host software and the embedded logic, contain a back-pressure scheme that makes both sides tolerant of overflow and starvation conditions, which is equivalent of USB 3.2 link layer. This path also uses native DCI packet protocol instead of USB 3.2 protocol. DCI.OOB - slower speed, CCA box needed. But survives S0ix and Sx states. Provides early boot access. Cannot tolerate re-driver circuits in its path.

30.6.2 USB 3.2 Hosting DCI.DBC

It relies on Debug Class Devices (DbC) which is comprised of a set of logic that is bolted to the side of the xHCI host controller and enable the target to act the role of a USB 3.2 device for debug purpose. This path uses the USB 3.2 packet protocol layer, USB 3.2 link layer flow control and USB 3.2 physical layer at 5 GHz. DCI.DBC - fast speed. USB 3.2 only works in S0. USB 2.0 survives S0ix and Sx states and provides early boot access and available on Ice Lake -UY B0

30.6.3 Platform Setup

Figure 30-2. Platform Setup with DCI Connection



§ §

31 Intel® Serial I/O Universal Asynchronous Receiver/ Transmitter (UART) Controllers

31.1 Acronyms

Acronyms	Description
DMA	Direct Memory Access
UART	Universal Asynchronous Receiver/Transmitter

31.2 Feature Overview

The PCH implements three independent UART interfaces, UART0, UART1 and UART2. Each UART interface is a 4-wire interface supporting up to 6.25 Mbit/s.

The interfaces can be used in the low-speed, full-speed, and high-speed modes. The UART communicates with serial data ports that conform to the RS-232 interface protocol.

UART2 only implements the UART Host controller and does not incorporate a DMA controller which is implemented for UART0 and UART1. Therefore, UART2 is restricted to operate in PIO mode only

The UART interfaces support the following features:

- Up to 6.25 Mbits/s Auto Flow Control mode as specified in the 16750 standard
- Transmitter Holding Register Empty (THRE) interrupt mode
- 64-byte TX and 64-byte RX host controller FIFOs
- DMA support with 64-byte DMA FIFO per channel (up to 32-byte burst)
- Functionality based on the 16550 industry standards
- Programmable character properties, such as number of data bits per character (5-8), optional parity bit (with odd or even select) and number of stop bits (1, 1.5, or 2)
- Line break generation and detection
- DMA signaling with two programmable modes
- Prioritized interrupt identification
- Programmable FIFO enable/disable
- Programmable serial data baud rate
- Modem and status lines are independently controlled
- Programmable BAUD RATE supported (baud rate = (serial clock frequency)/ (16xdivisor))

Notes:

1. SIR mode is not supported.
2. External read enable signal for RAM wake up when using external RAMs is not supported.

31.3 Signal Description

Name	Type	Description
UART0_RXD/ GPP_C8	I	**UART 0 Receive Data**
UART0A_TXD/ GPP_C9	O	**UART 0 Transmit Data**
UART0A_RTS#/ GPP_C10	O	**UART 0 Request to Send**
UART0A_CTS#/ GPP_C11	I	**UART 0 Clear to Send**
UART0B_RXD/ GPP_F1 / CNV_BRI_RSP	I	**Second Instant of UART 0 Receive Data**
UART0B_TXD/ GPP_F2 / CNV_RGI_DT	O	**Second Instant of UART 0 Transmit Data**
UART0B_RTS#/ GPP_F0 / CNV_BRI_DT	O	**Second Instant of UART 0 Request to Send**
UART0B_CTS#/ GPP_F3 / CNV_RGI_RSP	I	**Second Instant of UART 0 Clear to Send**
UART1_RXD/ ISH_UART1_RXD/ GPP_C12	I	**UART 1 Receive Data**
UART1_TXD/ ISH_UART1_TXD/ GPP_C13	O	**UART 1 Transmit Data**
UART1_RTS#/ ISH_UART1_RTS#/ GPP_C14	O	**UART 1 Request to Send**
UART1_CTS#/ ISH_UART1_CTS#/ GPP_C15	I	**UART 1 Clear to Send**
UART2_RXD/ GPP_C20	I	**UART 2 Receive Data**
UART2_TXD/ GPP_C21	O	**UART 2 Transmit Data**
UART2_RTS#/ GPP_C22	O	**UART 2 Request to Send**
UART2_CTS#/ GPP_C23	I	**UART 2 Clear to Send**

31.4 Integrated Pull-Ups and Pull-Downs

None

31.5 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S3/S4/S5	Deep Sx
UART[2:0]_RXD	Primary	Undriven	Undriven	Undriven	OFF
UART[2:0]_TXD	Primary	Undriven	Undriven	Undriven	OFF
UART[2:0]_RTS#	Primary	Undriven	Undriven	Undriven	OFF
UART[2:0]_CTS#	Primary	Undriven	Undriven	Undriven	OFF

Notes:
1. Reset reference for primary well pins is RSMRST#.

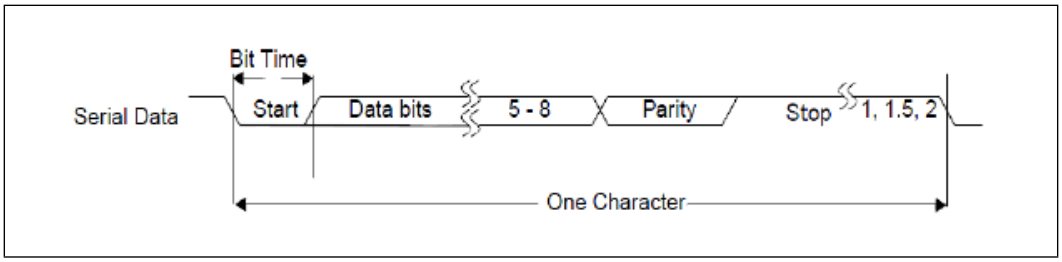
31.6 Functional Description

31.6.1 UART Serial (RS-232) Protocols Overview

Because the serial communication between the UART host controller and the selected device is asynchronous, Start and Stop bits are used on the serial data to synchronize the two devices. The structure of serial data accompanied by Start and Stop bits is referred to as a character.

An additional parity bit may be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure to provide the UART Host Controller with the ability to perform simple error checking on the received data.

Figure 31-1. UART Serial Protocol



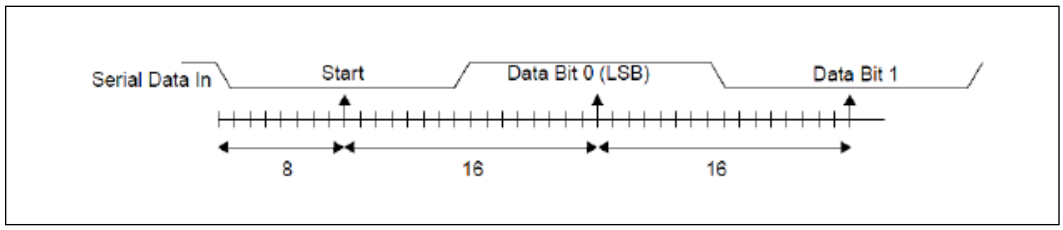
The UART Host Controller Line Control Register (LCR) is used to control the serial character characteristics. The individual bits of the data word are sent after the Start bit, starting with the least significant bit (LSB). These are followed by the optional parity bit, followed by the Stop bit(s), which can be 1, 1.5, or 2.

The Stop bit duration implemented by UART host controller may appear longer due to idle time inserted between characters for some configurations and baud clock divisor values in the transmit direction.

All bit in the transmission (with exception to the half stop bit when 1.5 stop bits are used) are transmitted for exactly the same time duration (which is referred to as Bit Period or Bit Time). One Bit Time equals to 16 baud clocks.

To ensure stability on the line, the receiver samples the serial input data at approximately the midpoint of the Bit Time once the start bit has been detected.

Figure 31-2. UART Receiver Serial Data Sample Points



31.6.2 16550 8-bit Addressing - Debug Driver Compatibility

Note: The PCH UART host controller is not compatible with legacy UART 16550 debug-port drivers. The UART host controller operates in 32-bit addressing mode only. UART 16550 legacy drivers only operate with 8-bit (byte) addressing. In order to provide compatibility with standard in-box legacy UART drivers a 16550 Legacy Driver mode has been implemented in the UART controller that will convert 8-bit addressed accesses from the 16550 legacy driver to the 32-bit addressing that the UART host controller supports.The UART 16550 8-bit Legacy mode only operates with PIO transactions. DMA transactions are not supported in this mode.

31.6.3 DMA Controller

The UART controllers 0 and 1 (UART0 and UART1) have an integrated DMA controller. Each channel contains a 64-byte FIFO. Max. burst size supported is 32 bytes.

UART controller 2 (UART2) only implements the host controllers and does not incorporate a DMA. Therefore, UART2 is restricted to operate in PIO mode only.

31.6.3.1 DMA Transfer and Setup Modes

The DMA can operate in the following modes:

1. Memory to peripheral transfers. This mode requires that the peripheral control the flow of the data to itself.
2. Peripheral to memory transfer. This mode requires that the peripheral control the flow of the data from itself.

The DMA supports the following modes for programming:

1. Direct programming. Direct register writes to DMA registers to configure and initiate the transfer.
2. Descriptor based linked list. The descriptors will be stored in memory (such as DDR or SRAM). The DMA will be informed with the location information of the descriptor. DMA initiates reads and programs its own register. The descriptors can form a linked list for multiple blocks to be programmed.
3. Scatter Gather mode.

31.6.3.2 Channel Control

- The source transfer width and destination transfer width are programmable. It can vary to 1 byte, 2 bytes, and 4 bytes.
- Burst size is configurable per channel for source and destination. The number is a power of 2 and can vary between 1,2,4,...,128. this number times the transaction width gives the number of bytes that will be transferred per burst.

- Individual Channel enables. If the channel is not being used, then it should be clock gated.
- Programmable Block size and Packing/Unpacking. Block size of the transfer is programmable in bytes. the block size is not be limited by the source or destination transfer widths.
- Address incrementing modes: The DMA has a configurable mechanism for computing the source and destination addresses for the next transfer within the current block. The DMA supports incrementing addresses and constant addresses.
- Flexibility to configure any hardware handshake sideband interface to any of the DMA channels.
- Early termination of a transfer on a particular channel.

31.6.4 Reset

Each host controller has an independent rest associated with it. Control of these resets is accessed through the Reset Register.

Each host controller and DMA will be in reset state once powered off and require SW (BIOS or driver) to write into specific reset register to bring the controller from reset state into operational mode.

31.6.5 Power Management

31.6.5.1 Device Power Down Support

In order to power down peripherals connected to PCH UART bus, the idle, configured state of the I/O signals must be retained to avoid transitions on the bus that can affect the connected powered peripheral. Connected devices are allowed to remain in the D0 active or D2 low power states when the bus is powered off (power gated). The PCH HW will prevent any transitions on the serial bus signals during a power gate event.

31.6.5.2 Latency Tolerance Reporting (LTR)

Latency Tolerance Reporting is used to allow the system to optimize internal power states based on dynamic data, comprehending the current platform activity and service latency requirements. The UART bus architecture, however, does not provide the architectural means to define dynamic latency tolerance messaging. Therefore, the interface supports this by reporting its service latency requirements to the platform power management controller via LTR registers.

The controller’s latency tolerance reporting can be managed by one of the two following schemes. The platform integrator must choose the correct scheme for managing latency tolerance reporting based on the platform, OS and usage.

1. Platform/HW Default Control. This scheme is used for usage models in which the controller’s state correctly informs the platform of the current latency requirements. In this scheme, the latency requirement is a function of the controller state. The latency for transmitting data to/from its connected device at a given rate while the controller is active is representative of the active latency requirements. On the other hand if the device is not transmitting or receiving data and idle, there is no expectation for end to end latency.
2. Driver Control. This scheme is used for usage models in which the controller state does not inform the platform correctly of the current latency requirements. If the

FIFOs of the connected device are much smaller than the controller FIFOs, or the connected device's end to end traffic assumptions are much smaller than the latency to restore the platform from low power state, driver control should be used.

31.6.6 Interrupts

UART interface has an interrupt line which is used to notify the driver that service is required.

When an interrupt occurs, the device driver needs to read both the host controller and DMA status and TX completion interrupt registers to identify the interrupt source. Clearing the interrupt is done with the corresponding interrupt register in the host controller or DMA.

All interrupts are active high and their behavior is level interrupt.

31.6.7 Error Handling

Errors that might occur on the external UART signals are comprehended by the host controller and reported to the interface host controller driver through the MMIO registers.

§ §

32 Universal Serial Bus (USB)

32.1 Acronyms

Acronyms	Description
xHCI	eXtensible Host Controller Interface

32.2 References

Specification	Location
USB 3.2 Specification	www.usb.org
USB 3.1 Specification	
USB 3.0 Specification	
USB 2.0 Specification	

32.3 Overview

The PCH implements an USB 3.2 xHCI controller which provides support for up to 10 USB 2.0 signal pairs and 6 USB 3.2 signal pairs. The xHCI controller supports wake up from sleep states S1-S4. The xHCI controller supports up to 64 devices for a max number of 2048 Asynchronous endpoints (Control / Bulk) or max number of 128 Periodic endpoints (Interrupt / isochronous).

Each walk-up USB 3.2 capable port must includes USB 3.2 and USB 2.0 signaling.

32.4 Signal Description

Table 32-1. Signal Descriptions (Sheet 1 of 3)

Name	Type	Description
USB31_1_RXN,/ USB31_1_RXP,/	I	**USB 3.2 Differential Receive Pair 1:** These are USB 3.2-based high-speed differential signals for Port #1 and the xHCI. It should be map to a USB connector with one of the OC (overcurrent) signals. ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 1.
USB31_1_TXN,/ USB31_1_TXP,/	O	**USB 3.2 Differential Transmit Pair 1:** These are USB 3.2-based high-speed differential signals for Port #1 and the xHCI. It should be map to a USB connector with one of the OC (overcurrent) signals. ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 1.
USB31_2_RXN,/ USB31_2_RXP,/	I	**USB 3.2 Differential Receive Pair 2:** These are USB 3.2-based high-speed differential signals for Port #2 and the xHCI. It should be map to a USB connector with one of the OC (overcurrent) signals. ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 2.
USB31_2_TXN,/ USB31_2_TXP,/	O	**USB 3.2 Differential Transmit Pair 2:** These are USB 3.2-based high-speed differential signals for Port #2 and the xHCI. It should be map to a USB connector with one of the OC (overcurrent) signals. ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 2.

Table 32-1. Signal Descriptions (Sheet 2 of 3)

Name	Type	Description
USB31_3_RXN,/ **USB31_3_RXP**,/	I	**USB 3.2 Differential Receive Pair 3:** These are USB 3.2-based high-speed differential signals for Port #3 and the xHCI. It should be map to a USB connector with one of the OC (overcurrent) signals. ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 3.
USB31_3_TXN,/ **USB31_3_TXP,/**	O	**USB 3.2 Differential Transmit Pair 3:** These are USB 3.2-based high-speed differential signals for Port #3 and the xHCI. It should be map to a USB connector with one of the OC (overcurrent) signals. ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 3.
USB31_4_RXN,/ **USB31_4_RXP,/**	I	**USB 3.2 Differential Receive Pair 4:** These are USB 3.2-based high-speed differential signals for Port #4 and the xHCI. It should be map to a USB connector with one of the OC (overcurrent) signals. ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 4.
USB31_4_TXN,/ **USB31_4_TXP,/**	O	**USB 3.2 Differential Transmit Pair 4:** These are USB 3.2-based high-speed differential signals for Port #4 and the xHCI. It should be map to a USB connector with one of the OC (overcurrent) signals. ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 4.
USB31_5_RXN/ PCIE5_RXN **USB31_5_RXP**/ PCIE5_RXP	I	**USB 3.2 Differential Receive Pair 5:** These are USB 3.2-based high-speed differential signals for Port #5 and the xHCI. It should be map to a USB connector with one of the OC (overcurrent) signals. ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 5.
USB31_5_TXN/ PCIE5_TXN **USB31_5_TXP/** PCIE5_TXP	O	**USB 3.2 Differential Transmit Pair 5:** These are USB 3.2-based high-speed differential signals for Port #5 and the xHCI. It should be map to a USB connector with one of the OC (overcurrent) signals. ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 5.
USB31_6_RXN/ PCIE6_RXN **USB31_6_RXP/** PCIE6_RXP	I	**USB 3.2 Differential Receive Pair 6:** These are USB 3.2-based high-speed differential signals for Port #6 and the xHCI. It should be map to a USB connector with one of the OC (overcurrent) signals. ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 6.
USB31_6_TXN/ PCIE6_TXN **USB31_6_TXP/** PCIE6_TXP	O	**USB 3.2 Differential Transmit Pair 6:** These are USB 3.2-based high-speed differential signals for Port #6 and the xHCI. It should be map to a USB connector with one of the OC (overcurrent) signals. ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 6.
USB2P_1, **USB2N_1**	I/O	**USB 2.0 Port 1 Transmit/Receive Differential Pair 1:** This USB 2.0 signal pair are routed to xHCI and should map to a USB connector with one of the OC (overcurrent) signals.
USB2P_2, **USB2N_2**	I/O	**USB 2.0 Port 2 Transmit/Receive Differential Pair 2:** This USB 2.0 signal pair are routed to xHCI and should map to a USB connector with one of the OC (overcurrent) signals.
USB2P_3, **USB2N_3**	I/O	**USB 2.0 Port 3Transmit/Receive Differential Pair 3:** This USB 2.0 signal pair are routed to xHCI and should map to a USB connector with one of the OC (overcurrent) signals.
USB2P_4, **USB2N_4**	I/O	**USB 2.0 Port 4 Transmit/Receive Differential Pair 4:** This USB 2.0 signal pair are routed to xHCI and should map to a USB connector with one of the OC (overcurrent) signals.
USB2P_5, **USB2N_5**	I/O	**USB 2.0 Port 5 Transmit/Receive Differential Pair 5:** This USB 2.0 signal pair are routed to xHCI and should map to a USB connector with one of the OC (overcurrent) signals.
USB2P_6, **USB2N_6**	I/O	**USB 2.0 Port 6 Transmit/Receive Differential Pair 6:** This USB 2.0 signal pair are routed to xHCI and should map to a USB connector with one of the OC (overcurrent) signals.
USB2P_7, **USB2N_7**	I/O	**USB 2.0 Port 7 Transmit/Receive Differential Pair 7:** This USB 2.0 signal pair are routed to xHCI and should map to a USB connector with one of the OC (overcurrent) signals.

Table 32-1. Signal Descriptions (Sheet 3 of 3)

Name	Type	Description
USB2P_8, USB2N_8	I/O	**USB 2.0 Port 8 Transmit/Receive Differential Pair 8:** This USB 2.0 signal pair are routed to xHCI and should map to a USB connector with one of the OC (overcurrent) signals.
USB2P_9, USB2N_9	I/O	**USB 2.0 Port 9 Transmit/Receive Differential Pair 9:** This USB 2.0 signal pair are routed to xHCI and should map to a USB connector with one of the OC (overcurrent) signals.
USB2P_10, USB2N_10	I/O	**USB 2.0 Port 10 Transmit/Receive Differential Pair 10:** This USB 2.0 signal pair are routed to xHCI and should map to a USB connector with one of the OC (overcurrent) signals.
USB_OC0# / GPP_E9	I	**Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred.
USB_OC1# / GPP_A14	I	**Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred.
USB_OC2#/	I	**Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred.
USB_OC3#/ GPP_A16	I	**Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred.
USB_VBUSSENSE	I	VBUS Sense for USB Device mode. Refer to OTG 3.0 specification for the sensing threshold voltage spec. ***Note:*** This HW signal is not used on the PCH for USB device mode functionality. This signal should be connected to ground.
USB_ID	I	ID detect for USB Device mode. ***Note:*** This HW signal is not used on the PCH for dual role mode selection. The switching of USB port role is done through the eSPI message from EC. This signal should be connected to ground.
USB2_COMP	I	USB Resistor Bias, analog connection points for an external resistor to ground.

32.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
USB2N_[10:1]	Internal Pull-down	14.25–24.8 kohm	1
USB2P_[10:1]	Internal Pull-down	14.25–24.8 kohm	1
USB_ID	Internal Weak Pull-up	14.25–24.8 kohm	If this signal is not in use, then the pin shall be connected directly to ground.
Note: 1. Series resistors (45 ohm ±10%)			

32.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[2]	Immediately After Reset[2]	S3/S4/S5	Deep Sx
USB31_[6:1]_RXN **USB31_[6:1]_RXP**	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
USB31_[6:1]_TXN **USB31_[6:1]_TXP**	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
USB2N_[10:1]	DSW	Internal Pull-down	Internal Pull-down	Internal Pull-down	Internal Pull-down
USB2P_[10:1]	DSW	Internal Pull-down	Internal Pull-down	Internal Pull-down	Internal Pull-down
USB_OC0#	Primary	Undriven	Undriven	Undriven	OFF
USB_OC1#	Primary	Undriven	Undriven	Undriven	OFF
USB_OC2#	Primary	Undriven	Undriven	Undriven	OFF
USB_OC3#	Primary	Undriven	Undriven	Undriven	OFF
USB_VBUSSENSE	Primary	Undriven	Undriven	Undriven	OFF
USB_ID[1]	Primary	Internal Pull-up	Undriven/ Internal Pull-up	Undriven/ Internal Pull-up	OFF
USB2_COMP	Primary	Undriven	Undriven	Undriven	OFF

Note:
1. The USB_ID pin is pulled-up internally.
2. Reset reference for primary well pins is RSMRST# and DSW well pins is DSW_PWROK.

32.7 Functional Description

32.7.1 eXtensible Host Controller Interface (xHCI) Controller

The eXtensible Host Controller Interface (xHCI) allows data transfer speed up to 10 Gb\s for USB 3.2 Gen 2x1 ports and 5 Gb\s for USB 3.2 Gen 1x1 ports. The xHCI supports SuperSpeed USB 10 Gbps, SuperSpeed USB 5 Gbps, High-Speed (HS), Full-Speed (FS) and Low-Speed (LS) traffic on the bus. The xHCI supports USB Debug port on all the USB ports. The xHCI also supports USB Attached SCIS Protocol (UASP).

32.7.2 USB Dual Role Support - eXtensible Device Controller Interface (xDCI) Controller

The USB subsystem also supports Dual Role Capability. The xHCI is paired with a standalone eXtensible Device Controller Interface (xDCI) to provide dual role functionality. The USB subsystem incorporates a xDCI USB 3.2 Gen 1x1 (5 Gb\s) device controller. The controller support SuperSpeed USB 5 Gbps on the CPU xDCI controller, and High-Speed (HS) on the PCH xDCI controller. These controllers are instantiated as a separate PCI function. The USB implementation is compliant to the Device specification and supports host/device only through the USB Type-C* connector.

The xDCI shares all USB ports with the host controller, with the ownership of the port being decided based the USB Power Delivery specification. Since all the ports support device mode, xDCI enabling must be extended by System BIOS and EC. While the port is mapped to the device controller, the host controller Rx detection must always indicate a disconnected port.

32.7.3 Supported USB 2.0 Ports

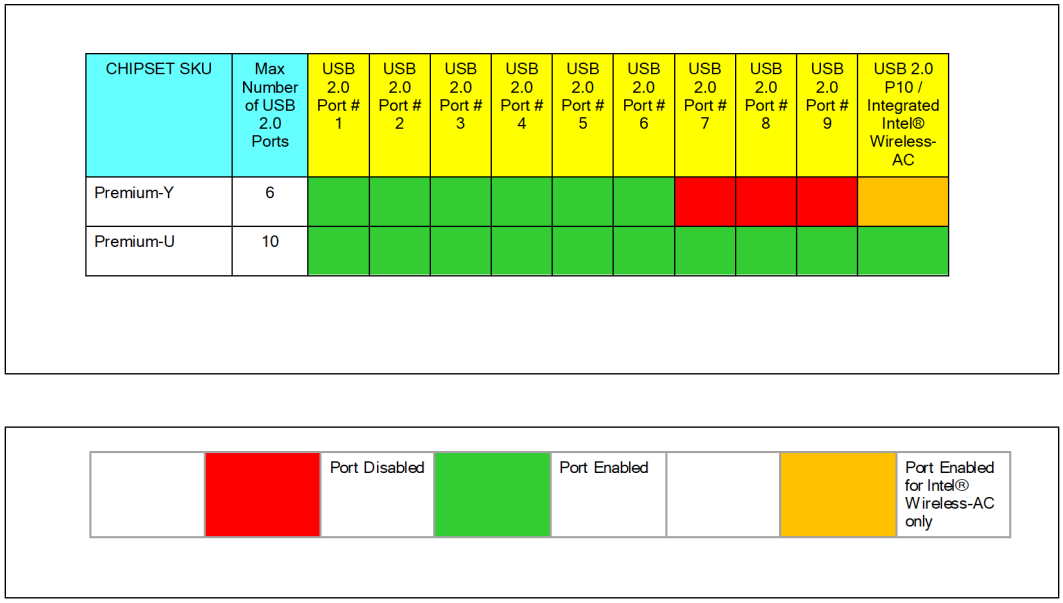
Due to the USB 2.0 port requirement for integrated Bluetooth® functionality with the integrated Intel® Wireless-AC (CNVi) solution, PCH-LP USB 2.0 port 10 will be available

- PCH-U
  - — USB 2.0 port 10 will be enabled on all platforms
- PCH-Y
  - — Not Applicable

If integrated Bluetooth® functionality is not desired, customers may utilize the port for USB functionality.

The total USB 2.0 port availability for a given SKU will also take into account the USB 2.0 port requirement for integrated Bluetooth® functionality. The following table describes the number of port supported and the associated port number enabled per SKU.

Figure 32-1. PCH-LP SKUs



§ §

33 Connectivity Integrated (CNVi)

33.1 Acronyms

Acronyms	Description
BRI	Bluetooth* Radio Interface
CNVi	Connectivity Integrated
PCH	Platform Controller Hub
RGI	Radio Generic interface
SoC	System On Chip
IP	Literally, Intellectual Property. IP refers to architecture, design, validation, and software components collectively delivered to enable one or more specific SoC features
MFUART	Multifunction Universal Asynchronous Receiver/Transmitter
UART	Universal Asynchronous Receiver/Transmitter

33.2 References

Specification	Location
M.2 Specification	https://pcisig.com/specifications/pciexpress/M.2_Specification/
MIPI® Alliance specification for D-PHY v1.2	https://mipi.org/specifications/

33.3 Overview

Connectivity Integrated (CNVi) is a general term referring to a family of connectivity solutions which are based on the Connectivity Controller family. The common component of all these solutions is the Connectivity Controller IP, which is a hard macro embedded in various Intel SoC chips.

The Integrated Connectivity (CNVi) solution consists of the following entities:

- The containing chip (SoC or PCH which contains the Connectivity Controller IP)
- Buttress (as applicable to each platform, and coupled the Connectivity Controller IP)
- Companion RF chip that is in a pre-certified module (i.e., M.2-2230, M.2-1216) or soldered as chip on board.

33.4 Signal Description

Name	Type	Description
GPIO fixed function		
GPP_A7 / I2S2_SCLK	I/O	For CNVi: Unused For standard CNV with UART host support: Optional Bluetooth I^2S bus clock
GPP_A8 / I2S2_SFRM / **CNV_RF_RESET#**	I/O	For CNVi: RF companion (CRF) reset signal, active low. Require a 75KOhm Pull-Down on platform/motherboard level. Recommended not use it for bootstrapping during early Platform init flows. For standard CNV with UART host support: Optional Bluetooth I^2S bus sync
GPP_A9 / I2S2_TXD/ **MODEM_CLKREQ**	O	For CNVi: Clock request signal. Used to request the RF companion clock (38.4M Ref clock) for CNL PCH; In PCH this function is not used, BUT this signal is also used for CNVi Init flow, so it must be connected on platform level even when clk sharing ability is not used/feasible. PCH is using internal clk (38.4 MHz clk) and **NOT taking this clk from the CRF** (as was optional in previous generations) For standard CNV with UART host Bluetooth* support: Optional Bluetooth* I^2S bus data output (input to BT module).
GPP_A10 / I2S2_RXD	I	For CNVi: Unused. For standard CNV with UART host support: Optional Bluetooth* I^2S bus data output (input to BT module) Note:
GPP_F0 / **CNV_BRI_DT** / UART0_RTS#	O	For CNVi: BRI bus TX. For standard CNV CNV with UART host support: BT UART RTS# Note: Require a 100-50-20KOhm (any of) Pull-up on platform/ motherboard level. Recommended not use it for bootstrapping during early Platform init flows
GPP_F1 / **CNV_BRI_RSP** / UART0_RXD	I	For CNVi: BRI bus RX. For standard CNV CNV with UART host support: BT UART RXD
GPP_F2 / **CNV_RGI_DT** / UART0_TXD	O	For CNVi: RGI bus TX. For standard CNV with UART host support: BT UART TXD
GPP_F3 / **CNV_RGI_RSP** / UART0_CTS#	I	For CNVi: RGI bus RX. For standard CNV with UART host support: BT UART CTS#
GPP_F4 / **CNV_RF_RESET#**	O	For CNVi (main): RF companion (CRF) reset signal, active low. Require a 75KOhm Pull-Down on platform/motherboard level. Recommended not use it for bootstrapping during early Platform Init flows.
GPP_F5 / **MODEM_CLKREQ**	O	For CNVi(main): Clock request signal. Used to request the RF companion clock (38.4M Ref clock) for CNL PCH; In PCH this function is not used, BUT this signal is also used for CNVi Init flow, so it must be connected on platform level even when clk sharing ability is not used/feasible. PCH using internal clk (38.4 MHz clk) and NOT taking this clk from the CRF (as was optional in previous generations)
GPP_F6 / **CNV_PA_BLANKING**	I	For CNVi and standard CNV: Optional WLAN/BT-WWAN coexistence signal COEX3. Used to be co-existence signal for external GNSS solution
GPP_H8 / I2C4_SDA / **CNV_MFUART2_RXD**	I/O	For CNVi and standard CNV: Optional WLAN/BT-WWAN coexistence signal COEX (Input)
GPP_H9 / I2C4_SCL / **CNV_MFUART2_TXD**	I/O	For CNVi and standard CNV: Optional WLAN/BT-WWAN coexistence signal COEX (Output)
Fixed special purpose I/O		
CNV_WT_CLKP	O	CNVio bus TX CLK+
CNV_WT_CLKN	O	CNVio bus TX CLK-
CNV_WT_D0P	O	CNVio bus Lane 0 TX+

Name	Type	Description
CNV_WT_D0N	O	CNVio bus Lane 0 TX-
CNV_WT_D1P	O	CNVio bus Lane 1 TX+
CNV_WT_D1N	O	CNVio bus Lane 1 TX-
CNV_WR_CLKP	I	CNVio bus RX CLK+
CNV_WR_CLKN	I	CNVio bus RX CLK-
CNV_WR_D0P	I	CNVio bus Lane 0 RX+
CNV_WR_D0N	I	CNVio bus Lane 0 RX-
CNV_WR_D1P	I	CNVio bus Lane 1 RX+
CNV_WR_D1N	I	CNVio bus Lane 1 RX-
CNV_WT_RCOMP	O	Wi-Fi DPHY RCOMP, analog connection point for an external bias resistor to ground
Selectable special purpose I/O		
USB2P_10		Bluetooth USB host bus (positive) for standard CNV. Optional to connect to a Bluetooth USB+ pin on the Bluetooth module. Port 10 is the recommended port but other USB 2.0 ports can be selected for this function.
USB2N_10		Bluetooth USB host bus (negative) for standard CNV. Optional to connect to a Bluetooth USB+ pin on the Bluetooth module. Port 10 is the recommended port but other USB 2.0 ports can be selected for this function.
PCIE14_TXP	O	Wi-Fi PCIe* host bus TX (positive) for standard CNV. Optional to connect to a Wi-Fi PCIe* PERp0 pin on the Wi-Fi module. This is the recommended port but other PCIe ports can be selected for this function.
PCIE14_TXN	O	Wi-Fi PCIe* host bus TX (negative) for standard CNV. Optional to connect to a Wi-Fi PCIe* PERn0 pin on the Wi-Fi module. This is the recommended port but other PCIe* ports can be selected for this function.
PCIE14_RXP	I	Wi-Fi PCIe* host bus RX (positive) for standard CNV. Optional to connect to a Wi-Fi PCIe* PETp0 pin on the Wi-Fi module. This is the recommended port but other PCIe* ports can be selected for this function.
PCIE14_RXN	I	Wi-Fi PCIe* host bus RX (negative) for standard CNV. Optional to connect to a Wi-Fi PCIe* PETn0 pin on the Wi-Fi module. This is the recommended port but other PCIe ports can be selected for this function.
CLKOUT_PCIE_P3	O	Wi-Fi PCIe* host bus clock (positive) for standard CNV. Optional to connect to a Wi-Fi PCIe* REFCLKp pin on the Wi-Fi module. This is the recommended clock signal but other PCIe* clocks can be selected for this function.
CLKOUT_PCIE_N3	O	Wi-Fi PCIe* host bus clock (negative) for standard CNV. Optional to connect to a Wi-Fi PCIe* REFCLKp pin on the Wi-Fi module. This is the recommended clock signal but other PCIe* clocks can be selected for this function.
CL_RST#	O	Wi-Fi CLINK host bus reset for standard CNV with CLINK support (Intel® vPro™). Optional to connect to a Wi-Fi CLINK reset pin on the Intel® vPro™ Wi-Fi* module.
CL_DATA	I/O	Wi-Fi CLINK host bus data for standard CNV with CLINK support (Intel® vPro™). Optional to connect to a Wi-Fi CLINK data pin on the Intel® vPro™ Wi-Fi module.
CL_CLK	O	Wi-Fi CLINK host bus clock for standard CNV with CLINK support (Intel® vPro™). Optional to connect to a Wi-Fi CLINK clock pin on the Intel® vPro™ Wi-Fi module.
W_disable1#(GPIO)	I	Used for Wi-Fi RF-Kill control. This pin can be connected to a platform switch or to SoC GPIOs (recommendation- if possible do not use GPIOs that have Platform impact as “bootstraps” during platform init).

Name	Type	Description
W_disable2#(GPIO)	I	Used for BT RF-Kill control. This pin can be connected to a platform switch or to SoC GPIOs (recommendation- if possible do not use GPIOs that have Platform impact as "bootstraps" during platform init).

33.5 Integrated Pull-ups and Pull-downs

Signal	Resistor	Value	Notes
CNV_BRI_RSP	Pull up	~120 kohm	Integrated in the CNVi IP/ Silicon
CNV_RGI_RSP	Pull up	~120 kohm	Integrated in the CNVi IP/ Silicon

33.6 Platform PU/PD Requirements

I/F	Signals	PU/PD in Platform	Comments
BRI/RGI Bluetooth * UART	CNV_RGI_DT	PU (20kohm)	This pull is required so that the SOC will be able to reliably detect that the CRF is present at power-up. However, it is possible to increase the resistor to 50K or even to 100K instead of 20K.
Init signals	RF_RESET_B	PD (75kohm)	It is highly encouraged to increase this resistor (or allow to switch it off when CNVi is active; not sure this is possible at the platform level). This resistor consumes power (43uW) all the time.
A4WP indication	A4WP_PRESENT (GPP_F19 in PCH-LP)	PD (75kohm)	Native function A4WP is not supported. The pin can instead be used as GPIO (when BIOS programs the pin to GPIO functionality). It is recommended to have an external pull down on the pin regardless of the pin being used or not to minimize power consumption. If the pin is used as GPIO, there should NOT be any on-board device driving the pin high until BIOS programs it to GPIO functionality.

33.7 I/O Signal Planes and States

Signal Name	Power plane	During Reset[1]	Immediately After Reset[1]	S3/S4/S5	Deep Sx
CNV_RF_RESET#	Primary	Driven	Driven	Driven	OFF
MODEM_CLKREQ	Primary	Driven	Driven	Driven	OFF
CNV_PA_BLANKING	Primary	Undriven	Undriven	Undriven	OFF
CNV_MFUART2_RXD	Primary	Undriven	Undriven	Undriven	OFF
CNV_MFUART2_TXD	Primary	Undriven	Undriven	Undriven	OFF
CNV_BRI_DT	Primary	Driven	Driven	Driven	OFF

Signal Name	Power plane	During Reset[1]	Immediately After Reset[1]	S3/S4/S5	Deep Sx
CNV_BRI_RSP	Primary	Powered (input, PU)	Powered (input, PU)	Powered (input, PU)	OFF
CNV_RGI_DT	Primary	Driven	Driven	Driven	OFF
CNV_RGI_RSP	Primary	Powered (input, PU)	Powered (input, PU)	Powered (input, PU)	OFF
CNV_WT_CLKP	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_CLKN	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_D0P	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_D0N	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_D1P	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_D1N	Primary	Undriven	Undriven	Driven	OFF
CNV_WR_CLKP	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_CLKN	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_D0P	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_D0N	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_D1P	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_D1N	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WT_RCOMP	Primary	Undriven	Undriven	Driven	OFF

Notes:
1. Reset reference for primary well pins is RSMRST#.

33.8 Functional Description

The main blocks of the integrated Connectivity solution are partitioned according to the following:

- **Connectivity Controller IP** contains:
  - Interfaces to the PCH
  - Debug and testing interfaces
  - Power management and clock Interfaces
  - Interface to the Companion RF module (CRF)
  - Interface to physical I/O pins controlled by the PCH
  - Interfaces to the LTE modem via PCH GPIO
- **Companion RF (CRF)**: This is the integrated connectivity M.2 module. The CRF Top contains:
  - Debug and testing interfaces
  - Power and clock Interfaces
  - Interface to the Connectivity Controller chip
- **Physical I/O pins**: The SCU units are responsible for generating and controlling the power and clock resources of Connectivity Controller and CRF. There are unique

SCUs in Connectivity Controller and CRF and their operation is coordinated due to power and clock dependencies. This coordination is achieved by signaling over a control bus (AUX) connecting Connectivity Controller and CRF.

Both Connectivity Controller and CRF have a dedicated AUX bus and arbiter. These two AUX buses are connected by a special interface that connects over the RGI bus. Each of the Connectivity Controller and CRF cores is dedicated to handle a specific connectivity function (Wi-Fi, Bluetooth).

Only the digital part of the connectivity function is located in Connectivity Controller cores, while the CRF cores handle some digital, but mostly analog and RF functionality. Each core in the Connectivity Controller has an interface to the host and an interface to its counterpart in CRF. CRF cores include an analog part which is connected to board level RF circuitry and to an antenna.

§ §

34 embedded Multi Media Card (e.MMC*)

The e.MMC* is a universal data storage and communication media. It is designed to cover a wide area of applications such as smart phones, tablets, computers, cameras, and so on. PCH supports only 1.8V operating devices and PCH supports e.MMC* version 5.1.

34.0.1 Key Features Supported

- HW Command Queuing support complaint to e.MMC* v5.1 specification
- Support enhanced Strobe for HS400 mode @1.8V
- Both ADMA2/DMA and Non-DMA mode of operation
- Transfers the data in 1 bit, 4 bit and 8 bit mode
- Support 64b address

34.1 Signals Description

Name	Type	Description
EMMC_CMD/GPP_F7	I/O	e.MMC* Command/Response
EMMC_DATA0/GPP_F8	I/O	e.MMC* Data 0
EMMC_DATA1/GPP_F9	I/O	e.MMC* Data 1
EMMC_DATA2/GPP_F10	I/O	e.MMC* Data 2
EMMC_DATA3/GPP_F11	I/O	e.MMC* Data 3
EMMC_DATA4/GPP_F12	I/O	e.MMC* Data 4
EMMC_DATA5/GPP_F13	I/O	e.MMC* Data 5
EMMC_DATA6/GPP_F14	I/O	e.MMC* Data 6
EMMC_DATA7/GPP_F15	I/O	e.MMC* Data 7
EMMC_RCLK/GPP_F16	I	e.MMC* Receive Clock
EMMC_CLK/GPP_F17	O	e.MMC* Clock
EMMC_RCOMP	I/O	e.MMC* compensation (200 Ohm +/- 1% pull down to ground)
EMMC_RESET#/GPP_F18	O	Reset (Recommend device reset to be connected to platform reset and not to this pin)

34.2 Integrated Pull-Ups and Pull-Downs

None.

34.3 I/O Signal Planes and States

Signal Name	Power Well	During Reset[1]	Immediately after Reset[1]	S0/S3/S4/S5	Deep Sx
EMMC_DATA[7:0]	Primary	Undriven	Undriven	Undriven	OFF
EMMC_RCLK	Primary	Undriven	Undriven	Undriven	OFF
EMMC_CLK	Primary	Undriven	Undriven	Undriven	OFF
EMMC_CMD	Primary	Undriven	Undriven	Undriven	OFF
EMMC_RCOMP	Primary	Undriven	Undriven	Undriven	OFF
EMMC_RESET#	Primary	Undriven	Undriven	Undriven	OFF

Notes:
1. Reset reference for primary well pins is RSMRST#.

Note: Internal weak pull resistor is default off but configurable (pu/pd/none) after boot. The pin may be "L" or "H" after reset depending on the configuration.

34.4 Functional Description

The Controller handles e.MMC* Protocol at transmission, packing data, adding cyclic redundancy check (CRC), start/end bit, and checking for transaction format correctness. Main supported features are listed below.

The e.MMC* main use case is to connect an on board external storage device.

34.4.1 e.MMC* 5.1 Command Queuing

Command Queuing (CQ) definition for e.MMC* includes new commands for issuing tasks to the device, for ordering the execution of previously issued tasks and for additional task management function. The host controller with CQ can queue up to 32 commands to the device and the device selects and indicates one of the queued commands to host for service.

The host controller implements additional logic for handling a door-bell based DMA for the 32 descriptor / task list and manages the entire CQ flow which includes:

- Fetch and send the tasks/commands to device using existing logic
- Maintains context of each queued command
- Periodically read the device queue status and indicates completion of task to SW.
- Implements interrupt coalescing to reduce burden on software ISR.

34.4.2 e.MMC* 5.1 Enhanced Strobe

Enhanced Strobe Mode for HS400 improves upon the HS400 mode interface speed increase that was first defined in e.MMC* version 5.0, by facilitating faster synchronization between the host and the device.

Refer JEDEC e.MMC* 5.1 specification for additional information.

34.4.3 e.MMC* Working Modes

The following table shows the working modes of e.MMC*.

Table 34-1. e.MMC* Working Modes

e.MMC* Mode	Data Rate	Clock Frequency	Max. Data Throughput
Compatibility	Single	0 – 26 MHz	26 MB/s
High Speed SDR	Single	0 – 52 MHz	52 MB/s
High Speed DDR	Dual	0 – 52 MHz	104MB/s
HS200	Single	0 - 200 MHz	200 MB/s
HS400	Dual	0 - 200 MHz	400 MB/s

§ §

35 Secure Digital eXtended Capacity (SDXC)

35.1 Acronyms

Acronyms	Description
SDXC	Secure Digital eXtended Capacity

35.2 References

Specification	Location
SDXC Specifications	http://www.sdcard.org

35.3 Feature Overview

The SDXC controller is to connect to an external detachable storage devices. It supports SDXC specification version 3.01.

35.4 Signal Description

Name	Type	Description	Voltage
SD_CMD /GPP_G0	I/O	SD Command/Response	3.3V or 1.8V
SD_DATA0/GPP_G1	I/O	SD Data	3.3V or 1.8V
SD_DATA1/GPP_G2	I/O	SD Data	3.3V or 1.8V
SD_DATA2/GPP_G3	I/O	SD Data	3.3V or 1.8V
SD_DATA3/GPP_G4	I/O	SD Data	3.3V or 1.8V
SD_CD#/GPP_G5	I	SDXC Detect	3.3V or 1.8V
SD_CLK/GPP_G6	O	SD clock	3.3V or 1.8V
SD_WP/GPP_G7	I	SDXC write protect	3.3V or 1.8V
SD_PWR_EN# / GPP_H1 / CNV_BT_I2S_SDO	O	SDXC power enable for 3.3V	1.8V
SD3_RCOMP	I/O	Impedance Compensation for 3.3V and 1,8V operation of SDXC and GPIO buffers. External reference resistor is required (200 Ohm ±1% pull down to ground) regardless of SDXC being used or not.	NA

35.5 Integrated Pull-Ups and Pull-Downs

None

35.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S3/S4/S5	Deep Sx
SD_CMD	Primary	Undriven	Undriven	Undriven	OFF
SD_DATA[3:0]	Primary	Undriven	Undriven	Undriven	OFF
SD_CD#	Primary	Undriven	Undriven	Undriven	OFF
SD_CLK	Primary	Undriven	Undriven	Undriven	OFF
SD_WP	Primary	Undriven	Undriven	Undriven	OFF
SD_PWR_EN#	Primary	Undriven	Driven	Undriven	OFF
SD3_RCOMP	Primary	Undriven	Undriven	Undriven	OFF
Notes: 1. Reset reference for primary well pins is RSMRST#.					

35.7 Functional Description

The SDXC controller handles SD Protocol at transmission, packing data, adding cyclic redundancy check (CRC), start/end bit, and checking for transaction format correctness. The SD Card main use case is to connect to an external detachable storage device. It supports SDXC card specification version 3.01. Both 1.8V and 3.3V signaling is supported. Additional information can be obtained from the SDXC 3.0 specification.

The following chart maps the working modes of SD Card.

Table 35-1. SD Working Modes

SD Card Mode	Data Rate	Clock Frequency	Maximum Data Throughput
Default Speed/SDR12	Single	0 – 25 MHz	12.5 MB/s
High Speed/SDR25	Single	0 – 50 MHz	25 MB/s
SDR50	Single	0 – 100 MHz	50 MB/s
DDR50	Dual	0 – 50 MHz	50 MB/s
SDR104	Single	0 – 208 MHz	104 MB/s

35.8 Virtual GPIO (vGPIO) for SD Card Implementation

vGPIO is a special type of GPIO implemented in the PCH for a specific functionality. vGPIO is not a physical GPIO; the signal is not balled out on the package. Programming the vGPIO is similar to programming a physical GPIO.

The PCH implements vGPIO39 (in GPIO community 1), which is specifically used for SD card detection as an interrupt generation. If the PCH integrated SD card is utilized, in conjunction of the SD_CD# pin to be used as card detect, a physical GPIO pin is required for interrupt generation. vGPIO39 is intended to replace the need for this addition physical GPIO if desired. SW needs to program the vGPIO accordingly to enable this functionality.

§ §

36 Private Configuration Space Target Port ID

The PCH incorporates a wide variety of devices and functions. The registers within these devices are mainly accessed through the primary interface, such as PCI configuration space and IO/MMIO space. Some devices also have registers that are distributed within the PCH Private Configuration Space at individual endpoints (Target Port IDs) which are only accessible through the PCH Sideband Interface.These PCH Private Configuration Space Registers can be addressed via SBREG_BAR or through SBI Index Data pair programming.

Table 36-1. Private Configuration Space Register Target Port IDs (Sheet 1 of 2)

PCH Device/Function Type	Target Port ID
OPI Configuration	88h
FIA Configuration	CFh
General Purpose I/O (GPIO) Community 0	6Eh
General Purpose I/O (GPIO) Community 1	6Dh
General Purpose I/O (GPIO) Community 2	6Ch
General Purpose I/O (GPIO) Community 4	6Ah
DCI	71h
PCIe Controller #1 (SPA)	80h
PCIe Controller #2 (SPB)	81h
PCIe Controller #3 (SPC)	82h
PCIe Controller #4 (SPD)	83h
SATA	D9h
SMBus	C6h
eSPI / SPI	72h
xHCI	70h
CNVi	73h
HSIO Strap Configuration	89h
PSF1	BAh
PSF2	BBh
PSF3	BCh
PSF4	BDh
PSF6	7Fh
PSF7	7Eh
PSF8	7Dh
ISH Controller	BEh
Real Time Clock (RTC)	C3h
Processor Interface, 8254 Timer, HPET, APIC	C4h
USB 2.0	CAh

Table 36-1. Private Configuration Space Register Target Port IDs (Sheet 2 of 2)

PCH Device/Function Type	Target Port ID
UART, I^2C, GSPI	CBh
Integrated Clock Controller (ICC)	DCh
CSI-2 Interface	A1h
General Purpose I/O (GPIO) Community 3	6Bh
eMMC/SDXC	C0h
PCIe Controller #5 (SPE)	84h
PCIe Controller #6 (SPF)	85h
USB Dual Role / OTG	E5h
MODPHY0	ABh
MODPHY1	AAh
MODPHY2	A9h
MODPHY3	A8h
Intel® Trace Hub	B6h

§ §