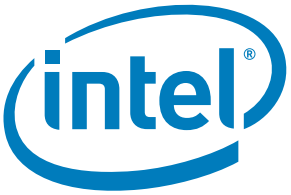


6th Generation Intel® Core™ Processor Families I/O Platform

Datasheet – Volume 1 of 2

Supporting 6th Generation Intel® Core™ Processor Family I/O based on U/Y-Processor Platforms

October 2020

Document Number: 332995-004EN

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.

Warning: Altering PC clock or memory frequency and/or voltage may (i) reduce system stability and use life of the system, memory and processor; (ii) cause the processor and other system components to fail; (iii) cause reductions in system performance; (iv) cause additional heat or other damage; and (v) affect system data integrity. Intel assumes no responsibility that the memory, included if used with altered clock frequencies and/or voltages, will be fit for any particular purpose. Check with memory manufacturer for warranty and additional details.

Code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.

I2C is a two-wire communications bus/protocol developed by NXP. SMBus is a subset of the I2C bus/protocol and was developed by Intel. Implementations of the I2C bus/protocol may require licenses from various entities, including NXP Semiconductors N.V.

Intel® Active Management Technology (Intel® AMT) requires activation and a system with a corporate network connection, an Intel® AMT-enabled chipset, network hardware and software. For notebooks, Intel AMT may be unavailable or limited over a host OS-based VPN, when connecting wirelessly, on battery power, sleeping, hibernating or powered off. Results dependent upon hardware, setup & configuration. For more information, visit http://www.intel.com/technology/platform-technology/intel-amt

Intel® High Definition Audio (Intel® HD Audio) Requires an Intel® HD Audio enabled system. Consult your PC manufacturer for more information. Sound quality will depend on equipment and actual implementation. For more information about Intel HD Audio, visit http://www.intel.com/design/chipsets/hdaudio.htm.

Intel® Rapid Storage Technology (Intel® RST) requires a select Intel® processor, enabled chipset, and Intel® Rapid Storage Technology (Intel® RST) software.

Intel® Smart Response Technology requires a Intel® Core™ processor, select Intel® chipset, Intel® Rapid Storage Technology software version 12.5 or higher, and a solid state hybrid drive reporting at least 16GB capacity and supporting SATA-I/O hybrid information feature. Depending on system configuration, your results may vary. Contact your system manufacturer for more information.

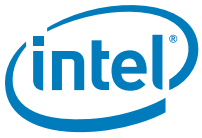
No computer system can provide absolute security under all conditions. Intel® Trusted Execution Technology (Intel® TXT) requires a computer system with Intel® Virtualization Technology, an Intel TXT-enabled processor, chipset, BIOS, Authenticated Code Modules and an Intel® TXT-compatible Measured Launched Environment (MLE). Intel® TXT also requires the system to contain a TPM v1.s. For more information, visit http://www.intel.com/technology/security.

Intel® Virtualization Technology requires a computer system with an enabled Intel processor, BIOS, virtual machine monitor (VMM). Functionality, performance or other benefits will vary depending on hardware and software configurations. Software applications may not be compatible with all operating systems. Consult your PC manufacturer. For more information, visit http://www.intel.com/go/virtualization

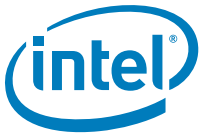
Intel, Intel® Active Management Technology (Intel® AMT), Intel® High Definition Audio (Intel® HD Audio), Intel® Rapid Storage Technology (Intel® RST), Intel® Smart Response Technology, Intel® Trusted Execution Technology (Intel® TXT), Intel® Virtualization Technology, Intel Core, and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by Intel Corporation is under license

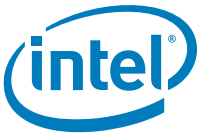
*Other names and brands may be claimed as the property of others.



Contents



Figures



Tables

Revision History

Document Number	Description	Date
004	Chapter 26, "Power Management" • Added note to SX_EXIT_HOLDOFF signal description	October 2020
003	Chapter 10, "Electrical Characteristics" • Updated Table 10-6	July 2019
002	Chapter 4, "Memory Mapping" • Removed Serial Port 3 in Table 4-3 Chapter 17, "General Purpose Input and Output (GPIO)" • Corrected native function for eSPI signals • Updated Table 17-2 Note column. Changed PLTRST# to PCH_PWROK for GPP_B14, GPP_B18, GPP_B22, GPP_E19, GPP_E21, GPP_E22, and GPP_E23 Chapter 22, "Integrated Sensor Hub (ISH)" • Typo correction on ISH UART 1 signals in Section 22.4 Chapter 31, "Serial Peripheral Interface (SPI)" • Updated PU/PD info for SPI CKL and CS# signals in section 31.5 and31.6 Chapter 38, "Secure Digital eXtended Capacity (SDXC)" • Clarified voltage on SD1.8_SEL and PWR_EN signals	May 2016
001	• Initial Release	February 2016

§ §

1 Introduction

1.1 About this Manual

This document is intended for Original Equipment Manufacturers (OEMs), Original Design Manufacturers (ODM) and BIOS vendors creating products based on the 6th Generation Intel® Core™ processor family I/O Platform Controller Hub (PCH).

Note: Throughout this document, the Platform Controller Hub (PCH) is used as a general term and refers to all 6th Generation Intel® Core™ Processor Family I/O PCH 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.

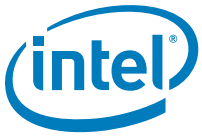
1.2 References

Specification	Document #/Location
6th Generation Intel® Core™ Processor Family I/O Platform Controller Hub Datasheet, Volume 2 of 2	332996

1.3 Overview

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

- ACPI Power Management Logic Support, Revision 4.0a
- PCI Express* Base Specification Revision 3.0
- Integrated Serial ATA Host controller, supports data transfer rates of up to 6Gb/s on all ports
- xHCI USB controller with SuperSpeed USB 3.0 ports
- USB Dual Role/OTG Capability
- MIPI*-Camera Serial Interface-2 (CSI-2)
- embedded MultiMedia Card (eMMC*) Revision 5.0 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 3.0
- General Purpose Input Output (GPIO)
- Low Pin Count (LPC) interface



- 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)
- 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

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

Table 1-1. PCH I/O Capabilities

Interface	PCH-Y	PCH-U
CPU Interface	OPI x8, 2GT/s and 4GT/s	OPI x8, 2GT/s and 4GT/s
PCIe*	Up to 10 Gen3 lanes (5 devices Max.)	Up to 12 Gen3 lanes (6 devices maximum)
USB	Up to 6 SS, 6 HS, 1 SSIC, 1 USB dual role ports	Up to 6 SS, 10 HS, 1 SSIC, 1 USB dual role ports
SATA	Up to 2 SATA Revision 3.0	Up to 3 SATA Revision 3.0
Camera	12 CSI-2 lanes	12 CSI-2 lanes
LAN Ports	1 GBE	1GBE
Audio	Intel® HD Audio, I^2S (Bluetooth®), Direct attach Digital MIC (DMIC)	Intel® HD Audio, I^2S (Bluetooth®), Direct attach Digital MIC (DMIC)
LPC	24 MHz, No DMA	24 MHz, No DMA
eSPI	1 CS#, Quad Mode	1 CS#, Quad Mode
I^2C	6	6
UART	3	3
Generic SPI (GSPI)	2	2
SDXC	SDXC3.0	SDXC3.0
Integrated Sensor Hub (ISH)	2 I^2C, 2 UART	2 I^2C, 2 UART
eMMC*	eMMC* 5.0	eMMC* 5.0

1.4 PCH SKUs

Table 1-2. PCH SKUs

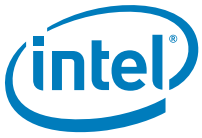
Features	Base-U	Premium-U	Premium-Y
Intel® Rapid Storage Technology (Intel® RST)	AHCI Mode	AHCI and RAID Mode	AHCI and RAID Mode
Total USB 3.0 Ports	4	6	6
Total USB 2.0 Ports	8[1]	10[2]	6[3]
Total SATA 3.0 Ports (Max. 6Gb/s)	2	3	2
Total PCI Express* Lanes (Gen)	10 (2.0)	12 (3.0)	10 (3.0)
Total Intel® RST for PCIe* Storage and SATA Express[4] Storage Devices	0	2[5]	2[5]
Total CSI2 lanes	8 lanes	12 lanes	12 lanes

Notes:
1. USB 2.0 port numbers: 1-8
2. USB 2.0 port numbers: 1-6
3. SATA Express Capable Ports (x2)
4. Intel® RST PCIe supports RAID configuration 0/1

Table 1-3. PCH HSIO Detail

SKU	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Base-U	USB 3.0/ OTG	USB 3.0/ SSIC	USB 3.0	USB 3.0	PCIe*	PCIe	PCIe/ LAN	PCIe/ LAN	PCIe/ LAN	PCIe	SATA	SATA	PCIe/ LAN	PCIe/ LAN	PCIe	PCIe
Premium-U	USB 3.0/ OTG	USB 3.0/ SSIC	USB 3.0	USB 3.0	PCIe* / USB 3.0	PCIe/ USB 3.0	PCIe/ LAN	PCIe/ LAN	PCIe/ LAN	PCIe	PCIe/ SATA	PCIe/ SATA	PCIe/ LAN	PCIe/ LAN	PCIe/ SATA	PCIe/ SATA
Premium-Y	USB 3.0/ OTG	USB 3.0/ SSIC	USB 3.0	USB 3.0	PCIe* / USB 3.0	PCIe/ USB 3.0	PCIe/ LAN	PCIe/ LAN	PCIe/ LAN	PCIe	PCIe/ SATA	PCIe/ SATA	PCIe/ LAN	PCIe/ LAN	N/A	N/A

§ §



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.

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

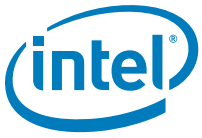
Device ID	Device Function - Device Description	C1 SRID	Notes
9D03	D23:F0 - SATA Controller (AHCI)	21	SKUs: PCH-U Base
9D07	D23:F0 - SATA Controller (AHCI, RAID)	21	3rd Party RAID [AIE=1]. SKUs: PCH-U/Y Premium, PCH-U Base
282A	D23:F0 - SATA Controller (RAID) Alternate ID	21	Intel® RST RAID [AIE=0, AIES=0]. SKUs: PCH-Y Premium, PCH-U Premium.
9D10	D28:F0 - PCI Express* Root Port #1	F1	
9D11	D28:F1 - PCI Express Root Port #2	F1	
9D12	D28:F2 - PCI Express Root Port #3	F1	
9D13	D28:F3 - PCI Express Root Port #4	F1	
9D14	D28:F4 - PCI Express Root Port #5	F1	
9D15	D28:F5 - PCI Express Root Port #6	F1	
9D16	D28:F6 - PCI Express Root Port #7	F1	
9D17	D28:F7 - PCI Express Root Port #8	F1	
9D18	D29:F0 - PCI Express Root Port #9	F1	
9D19	D29:F1 - PCI Express Root Port #10	F1	
9D1A	D29:F2 - PCI Express Root Port #11	F1	
9D1B	D29:F3 - PCI Express Root Port #12	F1	
9D20	D31:F1 – P2SB	21	
9D21	D31:F2 - Power Management Controller	21	
9D23	D31:F4 - SMBUS	21	
9D24	D31:F5 – SPI Controller	21	
9D25	D31:F6 – GbE Controller	21	
9D26	D31:F7 - Intel® Trace Hub	21	
9D27	D30:F0 – UART #0	21	See Note 1
9D28	D30:F1 – UART #1	21	See Note 1
9D29	D30:F2 – GSPI #0	21	See Note 1
9D2A	D30:F3 – GSPI #1	21	See Note 1
9D2B	D30:F4 – eMMC	21	See Note 1
9D2D	D30:F6 – SDXC	21	See Note 1

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

Device ID	Device Function - Device Description	C1 SRID	Notes
9D2F	D20:F0 – USB 3.0 xHCI Controller	21	
9D30	D20:F1 – USB Device Controller (OTG)	21	
9D31	D20:F2 – Thermal Subsystem	21	
9D32	D20:F3 – Camera IO Host Controller	21	
9D35	D19:F0 - ISH	21	
9D3A	D22:F0 - Intel® MEI #1	21	
9D3B	D22:F1 – Intel® MEI #2	21	
9D3C	D22:F2 – IDE Redirection	21	
9D3D	D22:F3 – Keyboard and Text (KT) Redirection	21	
9D3E	D22:F4 – Intel® MEI #3	21	
9D40-9D5F	D31:F0 - LPC or eSPI Controller	21	**PCH Device IDs:** Full featured engineering sample: 9D41 PCH-U Base: 9D43 PCH-Y Premium: 9D46 PCH-U Premium: 9D48
9D60	D21:F0 – I^2C Controller #0	21	
9D61	D21:F1 – I^2C Controller #1	21	
9D62	D21:F2 – I^2C Controller #2	21	
9D63	D21:F3 – I^2C Controller #3	21	
9D64	D25:F2 – I^2C Controller #4	21	
9D65	D25:F1 – I^2C Controller #5	21	
9D66	D25:F0 – UART Controller #2	21	
9D70	D31:F3 - Intel® High Definition Audio (Intel® HD Audio) (Audio, Voice, Speech)	21	

Note:
1. No more than 4 functions in Device 30 can be enabled in PCH.

§ §



3 Flexible I/O

3.1 Acronyms

Acronyms	Description
HSIO	High Speed I/O lanes
OTG	On-the-Go

3.2 References

None.

3.3 Overview

Flexible I/O is an architecture that allows some high-speed signals to be statically configured as PCI Express* (PCIe*), USB 3.0 or SATA signals per I/O needs on a platform.

3.4 Description

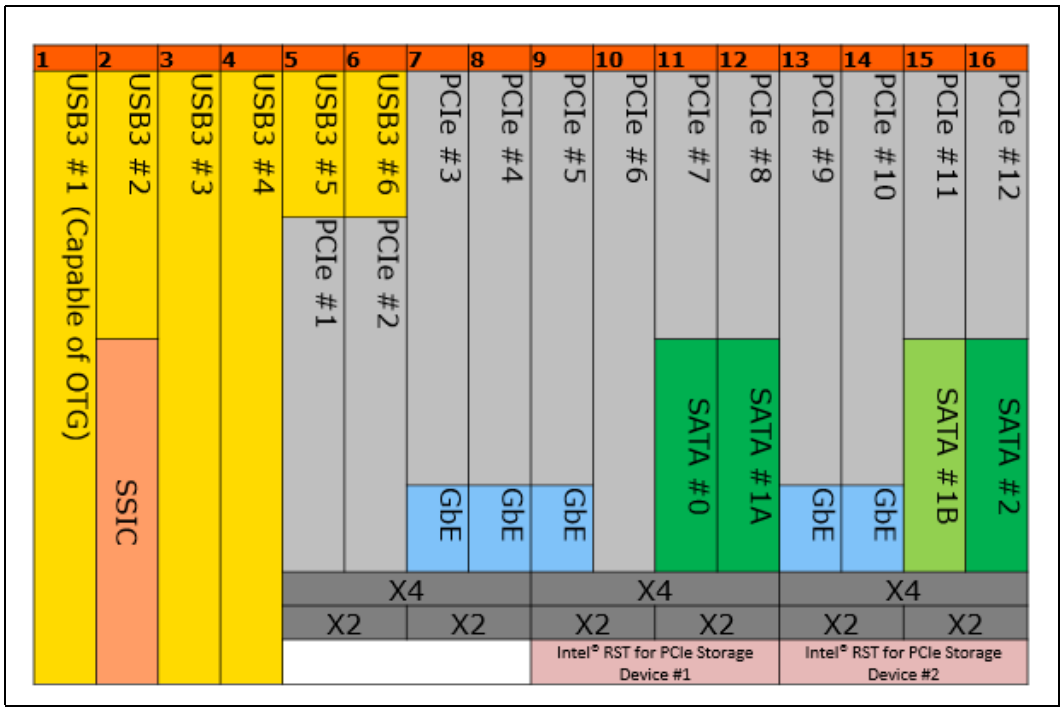
The PCH implements a number of high-speed I/O (HSIO) lanes that are split between the different interfaces, PCIe*, USB 3.0, SATA, GbE, USB Dual Role (OTG), and SSIC ports. The HSIO multiplexing is different between PCH-Y and PCH-U. Figure 3-1 and Figure 3-2 summarize the PCH HSIO lanes multiplexing.

The Flexible I/O is configured through soft straps.

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

3.4.1 PCH-U Flexible I/O

Figure 3-1. HSIO Multiplexing on PCH-U

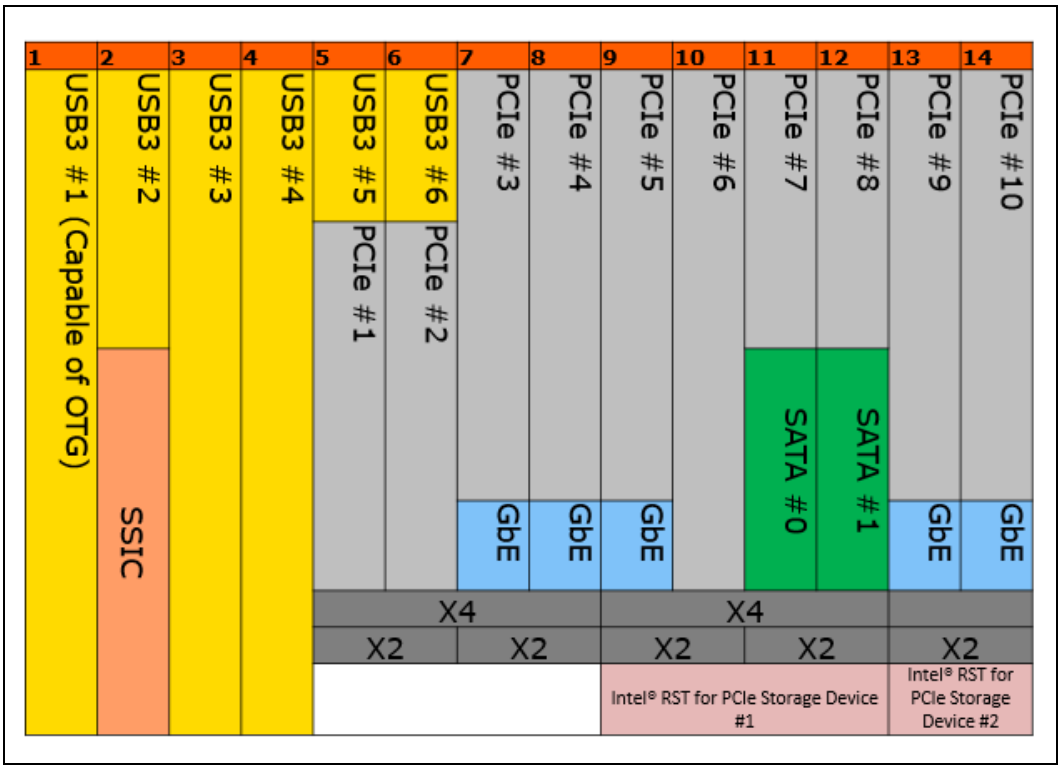


There are 16 HSIO lanes on the PCH-U, supporting the following port configurations:

1. Up to 12 PCIe* lanes (multiplexed with USB 3.0 ports, SATA Ports)
   - Only a maximum of 6 PCIe* ports (or devices) can be enabled at any time.
   - Ports 1-4, Ports 5-8, and Ports 9-12, can each be individually configured as 4x1, 2x2, 1x2 + 2x1, or 1x4.
2. Up to 3 SATA ports (multiplexed with PCIe*)
   - SATA Port 1 has the flexibility to be mapped to either PCIe* Port 8 or Port 11.
3. Up to 6 USB 3.0 ports (multiplexed with PCIe*)
   - USB Dual Role (OTG) capability is available on USB 3.0 Port 1
   - One SSIC x1 port is multiplexed with USB 3.0 Port 2
4. One GbE lane
   - GbE can be mapped into one of the PCIe* Ports 3-5 and Ports 9-10
   - When GbE is enabled, there can be at most up to 5 PCIe* ports enabled.
5. Up to 2 Intel RST for PCIe* storage devices supported
   - Devices can be x2 or x4
   - Devices can be implemented on PCIe Ports 5-8 and Ports 9-12

3.4.2 PCH-Y Flexible I/O

Figure 3-2. HSIO Multiplexing on PCH-Y



There are 14 HSIO lanes on the PCH-Y, supporting the following port configurations:

1. Up to 10 PCIe* lanes (multiplexed with USB 3.0 ports, SATA Ports)
   - Only a maximum of 6 PCIe* ports (or devices) can be enabled at any time
   - Ports 1-4, Ports 5-8, each can be individually configured as 4x1, 2x2, 1x2 + 2x1, or 1x4. Ports 9-10 can be configured as 2x1 or 1x2
2. Up to 2 SATA ports (multiplexed with PCIe*)
   - SATA Port 0 and 1 are muxed with PCIe* Port 7 and 8
3. Up to 6 USB 3.0 ports (multiplexed with PCIe*)
   - USB Dual Role (OTG) capability is available on USB 3.0 Port 1
   - One SSIC x1 port is multiplexed with USB 3.0 Port 2
4. One GbE lane
   - GbE can be mapped into one of the PCIe* Ports 3-5 and Ports 9-10
   - When GbE is enabled, there can be at most up to 5 PCIe* ports enabled
5. Up to 2 Intel RST for PCIe* storage devices supported
   - Devices can be x2 or x4.
   - Devices can be implemented on PCIe Ports 5-8 and Ports 9-10
   - Maximum number of devices that can be supported with RST are SKU-dependent.

3.5 HSIO Port Selection

The HSIO port configuration is statically selected by soft straps.

3.5.1 PCIe/SATA Port Selection

In addition to static configuration using soft straps, HSIO lanes that have PCIe/SATA port multiplexing can be configured using 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.

§ §

4 Memory Mapping

4.1 Overview

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

4.2 Functional Description

4.2.1 PCI Devices and Functions

The PCH incorporates a variety of PCI devices and functions, as shown in Table 4-1. 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 (See Chapter 20, "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)

Device: Functions #	Function Description
Bus 0: Device 31: Function 0	LPC Interface (eSPI Enable Strap = 0) eSPI Interface (eSPI Enable Strap = 1)
Bus 0: Device 31: Function 1	P2SB
Bus 0: Device 31: Function 2	PMC
Bus 0: Device 31: Function 3	Intel® High Definition Audio (Intel® HD Audio) (Audio, Voice, Speech)
Bus 0: Device 31: Function 4	SMBus Controller
Bus 0: Device 31: Function 5	SPI
Bus 0: Device 31: Function 6	GbE Controller
Bus 0: Device 31: Function 7	Intel® Trace Hub
Bus 0: Device 30: Function 0	UART #0
Bus 0: Device 30: Function 1	UART #1
Bus 0: Device 30: Function 2	SPI #0
Bus 0: Device 30: Function 4	eMMC*
Bus 0: Device 30: Function 6	SDCard
Bus 0: Device 29: Function 0	PCI Express Port 9
Bus 0: Device 29: Function 1	PCI Express Port 10
Bus 0: Device 29: Function 2	PCI Express Port 11
Bus 0: Device 29: Function 3	PCI Express Port 12
Bus 0: Device 28: Function 0	PCI Express Port 1
Bus 0: Device 28: Function 1	PCI Express Port 2

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

Device: Functions #	Function Description
Bus 0: Device 28: Function 2	PCI Express Port 3
Bus 0: Device 28: Function 3	PCI Express Port 4
Bus 0: Device 28: Function 4	PCI Express Port 5
Bus 0: Device 28: Function 5	PCI Express Port 6
Bus 0: Device 28: Function 6	PCI Express Port 7
Bus 0: Device 28: Function 7	PCI Express Port 8
Bus 0: Device 25: Function 0	UART Controller #2
Bus 0: Device 25: Function 1	I^2C Controller #5
Bus 0: Device 25: Function 2	I^2C Controller #4
Bus 0: Device 23: Function 0	SATA Controller
Bus 0: Device 22: Function 0	Intel® MEI #1
Bus 0: Device 22: Function 1	Intel® MEI #2
Bus 0: Device 22: Function 2	IDE Redirection (IDE-R)
Bus 0: Device 22: Function 3	Keyboard and Text (KT) Redirection
Bus 0: Device 22: Function 4	Intel® MEI #3
Bus 0: Device 21: Function 0	I^2C Controller #0
Bus 0: Device 21: Function 1	I^2C Controller #1
Bus 0: Device 21: Function 2	I^2C Controller #2
Bus 0: Device 21: Function 3	I^2C Controller #3
Bus 0: Device 20: Function 0	USB 3.0 xHCI Controller
Bus 0: Device 20: Function 1	USB Device Controller (OTG)
Bus 0: Device 20: Function 2	Thermal Subsystem
Bus 0: Device 20: Function 3	Camera I/O Host Controller
Bus 0: Device 19: Function 0	Integrated Sensor Hub
Note: When a device or function is disabled, it is not reported to the software and will not respond to any register reads or writes.	

4.2.2 Fixed I/O Address Ranges

Table 4-2, "Fixed I/O Ranges Decoded by PCH" shows the Fixed I/O decode ranges from the processor perspective. Note that 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.

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	Enable/Disable
20h – 21h	Interrupt Controller	Interrupt Controller	Interrupt	None
24h – 25h	Interrupt Controller	Interrupt Controller	Interrupt	None

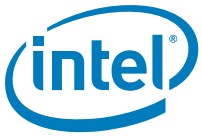


Table 4-2. Fixed I/O Ranges Decoded by PCH (Sheet 2 of 3)

I/O Address	Read Target	Write Target	Internal Unit	Enable/Disable
28h – 29h	Interrupt Controller	Interrupt Controller	Interrupt	None
2Ch – 2Dh	Interrupt Controller	Interrupt Controller	Interrupt	None
2Eh – 2Fh	LPC/eSPI	LPC/eSPI	Forwarded to LPC/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
4Eh – 4Fh	LPC/eSPI	LPC/eSPI	Forwarded to LPC/eSPI	Yes IOE.ME2
50h	Timer/Counter	Timer/Counter	8254 Timer	None
52h – 53h	Timer/Counter	Timer/Counter	8254 Timer	None
60h	LPC/eSPI	LPC/eSPI	Forwarded to LPC/eSPI	Yes w/ 60h IOE.KE
61h	NMI Controller	NMI Controller	Processor I/F	None
62h	Microcontroller	Microcontroller	Forwarded to LPC/eSPI	Yes w/ 66h IOE.ME1
63h	NMI Controller[1]	NMI Controller[1]	Processor I/F	Yes, alias to 61h GCS.P61AE
64h	Microcontroller	Microcontroller	Forwarded to LPC/eSPI	Yes w/ 60h and IOE.KE
65h	NMI Controller[1]	NMI Controller[1]	Processor I/F	Yes, alias to 61h GCS.P61AE
66h	Microcontroller	Microcontroller	Forwarded to LPC/eSPI	Yes w/ 62h IOE.ME1
67h	NMI Controller[1]	NMI Controller[1]	Processor I/F	Yes, alias to 61h GCS.P61AE
70h	RTC Controller	NMI and RTC Controller	RTC	None
71h	RTC Controller	RTC Controller	RTC	None
72h	RTC Controller	RTC Controller	RTC	Yes, w/ 72h RC.UE
73h	RTC Controller	RTC Controller	RTC	Yes, w/ 73h RC.UE
74h	RTC Controller	RTC Controller	RTC	None
75h	RTC Controller	RTC Controller	RTC	None
76h – 77h	RTC Controller	RTC Controller	RTC	Yes RC.UE
80h	LPC/eSPI or PCIe	LPC/eSPI or PCIe	LPC/eSPI or PCIe	GCS.RPR

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

I/O Address	Read Target	Write Target	Internal Unit	Enable/Disable
84h – 86h	Reserved	LPC/eSPI or PCIe	LPC/eSPI or PCIe	GCS.RPR
88h	Reserved	LPC/eSPI or PCIe	LPC/eSPI or PCIe	GCS.RPR
8Ch – 8Eh	Reserved	LPC/eSPI or PCIe	LPC/eSPI or PCIe	GCS.RPR
90h	(Alias to 80h)	(Alias to 80h)	Forwarded to LPC/eSPI	Yes, alias to 80h
92h	Reset Generator	Reset Generator	Processor I/F	None
94h – 96h	(Aliases to 8xh)	(Aliases to 8xh)	Forwarded to LPC/eSPI	Yes, aliases to 8xh
98h	(Alias to 88h)	(Alias to 88h)	Forwarded to LPC/eSPI	Yes, alias to 88h
9Ch – 9Eh	(Alias to 8xh)	(Aliases to 8xh)	Forwarded to LPC/eSPI	Yes, aliases 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 LPC/eSPI	Yes IOE.LGE
208–20Fh	Gameport High	Gameport High	Forwarded to LPC/eSPI	Yes IOE.HGE
4D0h – 4D1h	Interrupt Controller	Interrupt Controller	Interrupt Controller	None
CF9h	Reset Generator	Reset Generator	Interrupt controller	None

Note:
1. Only if the Port 61 Alias Enable bit (GCS.P61AE) bit is set. Otherwise, the target is PCIe*.

4.2.3 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 config 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.

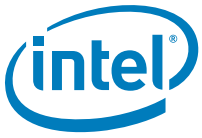


Table 4-3. Variable I/O Decode Ranges

Range Name	Mappable	Size (Bytes)	Target
ACPI	Anywhere in 64K I/O Space	96	Power Management
Primary IDE Bus	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	LPC Peripheral
Serial Port 1	8 Ranges in 64K I/O Space	8	LPC Peripheral
Serial Port 2	8 Ranges in 64K I/O Space	8	LPC Peripheral
Floppy Disk Controller	2 Ranges in 64K I/O Space	8	LPC Peripheral
LPC Generic 1	Anywhere in 64K I/O Space	4 to 256 bytes	LPC/eSPI
LPC Generic 2	Anywhere in 64K I/O Space	4 to 256 bytes	LPC/eSPI
LPC Generic 3	Anywhere in 64K I/O Space	4 to 256 bytes	LPC/eSPI
LPC Generic 4	Anywhere in 64K I/O Space	4 to 256 bytes	LPC/eSPI
I/O 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 1-12
Keyboard and Text (KT)	Anywhere in 64K I/O Space	8	Intel® AMT Keyboard and Text Redirection
Note: All ranges are decoded directly from DMI.			

4.3 Memory Map

Table 4-4, "PCH Memory Decode Ranges (Processor Perspective)" 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 primary aborted.

PCIe* cycles generated by external primary PCIe* 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
000E0000 – 000EFFFF	LPC/eSPI or SPI	Bit 6 in BIOS Decode Enable Register is set
000F0000 – 000FFFFF	LPC/eSPI or SPI	Bit 7 in BIOS Decode Enable Register is set
FECXX000 – FECXX040	I/O(x) APIC inside PCH	X controlled via APIC Range Select (ASEL) field and Enable (AEN) bit.
FEC10000 – FEC17FFF	PCIe* port 1	PCIe root port 1 APIC Enable (PAE) set
FEC18000 – FEC1FFFF	PCIe* port 2	PCIe root port 2 APIC Enable (PAE) set
FEC20000 – FEC27FFF	PCIe* port 3	PCIe root port 3 APIC Enable (PAE) set
FEC28000 – FEC2FFFF	PCIe* port 4	PCIe* root port 4 APIC Enable (PAE) set

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

Memory Range	Target	Dependency/Comments
FEC30000 – FEC37FFF	PCIe* port 5	PCIe* root port 5 APIC Enable (PAE) set
FEC38000 – FEC3FFFF	PCIe* port 6	PCIe* root port 6 APIC Enable (PAE) set
FEC40000 – FEC47FFF	PCIe* port 7	PCIe* root port 7 APIC Enable (PAE) set
FEC48000 – FEC4FFFF	PCIe* port 8	PCIe* root port 8 APIC Enable (PAE) set
FEC50000 – FEC57FFF	PCIe* port 9	PCIe* root port 9 APIC Enable (PAE) set
FEC58000 – FEC5FFFF	PCIe* port 10	PCIe* root port 10 APIC Enable (PAE) set
FEC60000 – FEC67FFF	PCIe* port 11	PCIe* root port 11 I/OxApic Enable (PAE) set
FEC68000 – FEC6FFFF	PCIe* port 12	PCIe* root port 12 I/OxApic Enable (PAE) set
FFC0 0000 – FFC7 FFFF FF80 0000 – FF87 FFFF	LPC/eSPI or SPI	Bit 8 in BIOS Decode Enable Register
FFC8 0000 – FFCF FFFF FF88 0000 – FF8F FFFF	LPC/eSPI or SPI	Bit 9 in BIOS Decode Enable Register
FFD0 0000 – FFD7 FFFF FF90 0000 – FF97 FFFF	LPC/eSPI or SPI	Bit 10 in BIOS Decode Enable Register is set
FFD8 0000 – FFDF FFFF FF98 0000 – FF9F FFFF	LPC/eSPI or SPI	Bit 11 in BIOS Decode Enable Register is set
FFE0 000 – FFE7 FFFF FFA0 0000 – FFA7 FFFF	LPC/eSPI or SPI	Bit 12 in BIOS Decode Enable Register is set
FFE8 0000 – FFEF FFFF FFA8 0000 – FFAF FFFF	LPC/eSPI or SPI	Bit 13 in BIOS Decode Enable Register is set
FFF0 0000 – FFF7 FFFF FFB0 0000 – FFB7 FFFF	LPC/eSPI or SPI	Bit 14 in BIOS Decode Enable Register is set
FFF8 0000 – FFFF FFFF FFB8 0000 – FFBF FFFF	LPC/eSPI or SPI	Always enabled. The top two 64-KB blocks in this range can be swapped by the PCH.
FF70 0000 – FF7F FFFF FF30 0000 – FF3F FFFF	LPC/eSPI or SPI	Bit 3 in BIOS Decode Enable Register is set
FF60 0000 – FF6F FFFF FF20 0000 – FF2F FFFF	LPC/eSPI or SPI	Bit 2 in BIOS Decode Enable Register is set
FF50 0000 – FF5F FFFF FF10 0000 – FF1F FFFF	LPC/eSPI or SPI	Bit 1 in BIOS Decode Enable Register is set
FF40 0000 – FF4F FFFF FF00 0000 – FF0F FFFF	LPC/eSPI or SPI	Bit 0 in BIOS Decode Enable Register is set
FED0 X000h – FED0 X3FFh	HPET	BIOS determines “fixed” location which is one of four 1-KB ranges where X (in the first column) is 0h, 1h, 2h, or 3h
FED4_0000h – FED4_7FFFh	LPC or SPI (set by strap)	TPM and Trusted Mobile KBC
FED5_0000h – FED5_FFFFh	Intel® ME	Always enabled
64 KB anywhere in 64-bit address range	USB 3.0 Host Controller	Enable via standard PCI mechanism (Device 20, Function 0)
2 MB anywhere in 4-Gb range	OTG	Enable via standard PCI mechanism (Device 20, Function 1)
24 KB anywhere in 4-Gb range	OTG	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)

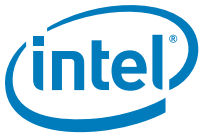


Table 4-4. PCH Memory Decode Ranges (Processor Perspective) (Sheet 3 of 3)

Memory Range	Target	Dependency/Comments
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)
64 KB anywhere in 4-GB range	LPC/eSPI	LPC Generic Memory Range. Enable via setting bit[0] of the LPC Generic Memory Range register (D31:F0:offset 98h)
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 1-12	Enable via standard PCI mechanism
Prefetchable Memory Base/ Limit anywhere in 64-bit address range	PCI Express Root Ports 1-12	Enable via standard PCI mechanism
4 KB anywhere in 64-bit address range	Thermal Reporting	Enable via standard PCI mechanism (Device 20: Function 2)
16 bytes anywhere in 64-bit address range	Intel® MEI#1, #2, #3,	Enable via standard PCI mechanism (Device 22: Function 0-1, 4)
4 KB anywhere in 4-GB range	Intel® AMT Keyboard and Text Redirection	Enable via standard PCI mechanism (Device 22: Function 3)
Twelve 4-KB slots anywhere in 64-bit address range	Intel Serial Interface controllers	Enable via standard PCI mechanism (Device 30: Function[7:0], Device 21: Function [6:0]
64 KB anywhere in 64-bit address range	Camera	Enable via standard PCI mechanism (Device 20: Function 3)
1 MB (BAR0) or 4 KB (BAR1) in 4-GB range	Integrated Sensor Hub	Enable via standard PCI mechanism (Device 19: Function 0)

4.3.1 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.

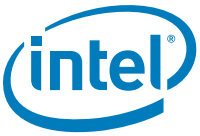
For FHW when top swap is enabled, accesses to FFFF_0000h-FFFF_FFFFh are directed to FFFE_0000h-FFFE_FFFFh and vice versa. When the Top Swap Enable bit is 0, the PCH will not invert A16.

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. SPI Mode Address Swapping

BOOT_BLOCK_SIZE Value	Accesses to	Being Directed to
000 (64 KB)	FFFF_0000h - FFFF_FFFFh	FFFE_0000h - FFFE_FFFFh and vice versa
001 (128 KB)	FFFE_0000h - FFFF_FFFFh	FFFC_0000h - FFFD_FFFFh and vice versa
010 (256 KB)	FFFC_0000h - FFFF_FFFFh	FFF8_0000h - FFFB_FFFFh and vice versa
011 (512 KB)	FFF8_0000h - FFFF_FFFFh	FFF0_0000h - FFF7_FFFFh and vice versa
100 (1 MB)	FFF0_0000h - FFFF_FFFFh	FFE0_0000h - FFEF_FFFFh and vice versa
Note: When the Top Swap Enable bit is 0, the PCH will not invert any address bit. This bit is automatically set to 0 by RTCRST#, but not by PLTRST#.		

§ §



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 References

None.

5.3 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 microcontroller.

5.4 Features

The following features and functions are supported by the PCH:

- First timer time-out 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 timeouts 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
- Processor present detection:
  - Detects if processor fails to fetch the first instruction after reset
- Various Error detections (such as ECC Errors) indicated by host controller:
  - Can generate SMI#, SCI, SERR, NMI, or TCO interrupt
- Intruder Detect input:
  - Can generate TCO interrupt or SMI# when the system cover is removed
  - INTRUDER# allowed to go active in any power state, including G3
- Detection of bad BIOS Flash programming:
  - Detects if data on first read is FFh (indicates that BIOS flash is not programmed)

5.4.1 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.

5.4.1.1 Detecting a System Lockup

When the processor is reset, it is expected to fetch its first instruction. If the processor fails to fetch the first instruction after reset, the TCO timer times out twice and the PCH asserts PLTRST#.

5.4.1.2 Handling an Intruder

The PCH has an input signal, INTRUDER#, that can be attached to a switch that is activated by the system's case being open. This input has a two RTC clock debounce. If INTRUDER# goes active (after the debouncer), this will set the INTRD_DET bit in the TCO2_STS register. The INTRD_SEL bits in the TCO_CNT register can enable the PCH to cause an SMI# or interrupt. The BIOS or interrupt handler can then cause a transition to the S5 state by writing to the SLP_EN bit.

The software can also directly read the status of the INTRUDER# signal (high or low) by clearing and then reading the INTRD_DET bit. This allows the signal to be used as a GPI if the intruder function is not required.

If the INTRUDER# signal goes inactive some point after the INTRD_DET bit is written as a 1, then the INTRD_DET bit will go to a 0 when INTRUDER# input signal goes inactive.

Note: This is slightly different than a classic sticky bit, since most sticky bits would remain active indefinitely when the signal goes active and would immediately go inactive when a 1 is written to the bit.

Note: The INTRD_DET bit resides in the PCH's RTC well, and is set and cleared synchronously with the RTC clock. Thus, when software attempts to clear INTRD_DET (by writing a 1 to the bit location) there may be as much as two RTC clocks (about 65 µs) delay before the bit is actually cleared. Also, the INTRUDER# signal should be asserted for a minimum of 1 ms to ensure that the INTRD_DET bit will be set.

Note: If the INTRUDER# signal is still active when software attempts to clear the INTRD_DET bit, the bit remains set and the SMI is immediately generated again. The SMI handler can clear the INTRD_SEL bits to avoid further SMIs. However, if the INTRUDER# signal goes inactive and then active again, there will not be further SMIs, since the INTRD_SEL bits would select that no SMI# be generated.

5.4.1.3 Detecting Improper Flash Programming

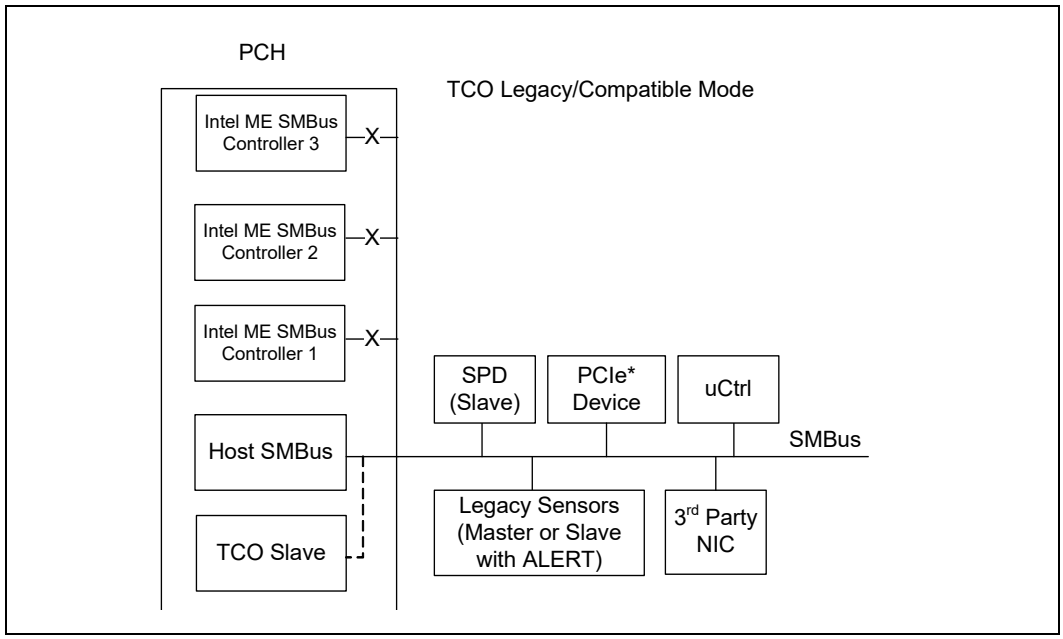
The PCH can detect the case where the BIOS flash is not programmed. This results in the first instruction fetched to have a value of FFh. If this occurs, the PCH sets the BAD_BIOS bit.

5.4.2 TCO Modes

5.4.2.1 TCO Compatible Mode

In TCO Legacy/Compatible mode, only the host SMBus is used. The Secondary TCO is connected to the host SMBus internally by default. In this mode, the Intel® Management Engine (Intel® ME) 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. Table 5-1 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
INTRUDER# pin	Yes	No	Must be in "hung S0" state
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

5.4.2.2 Advanced TCO Mode

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

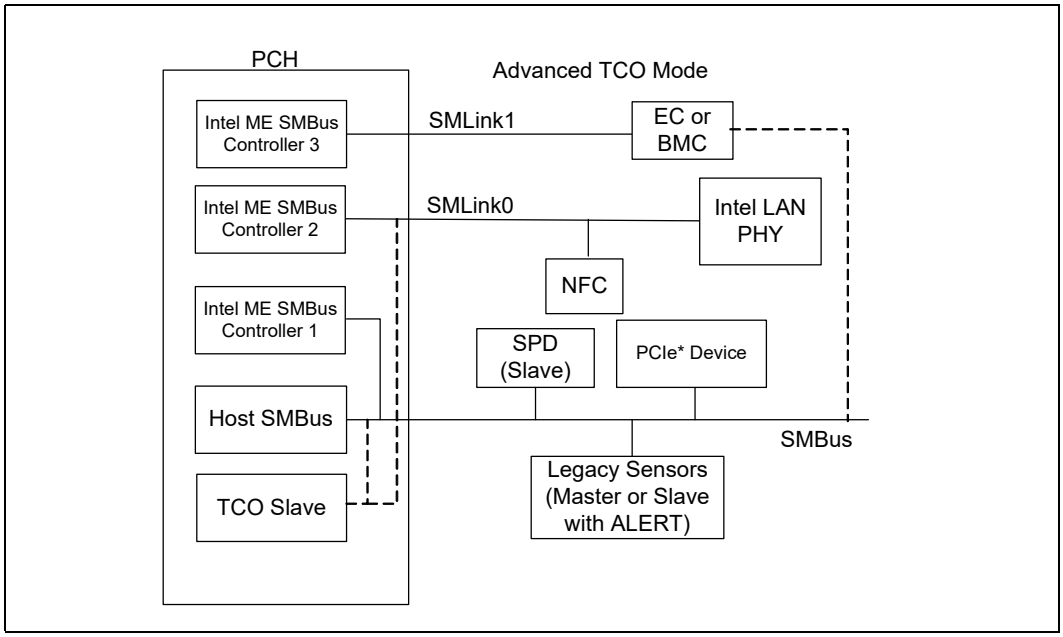
In this mode, the Intel® ME SMBus controllers must be enabled by soft strap in the flash descriptor. See Figure 5-2 for more details.

In advanced TCO mode, the secondary TCO 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® ME SMBus connected to SMLink1. The host and secondary TCO communicate with BMC through SMBus.

Figure 5-2. Advanced TCO Mode



§ §

6 High Precision Event Timer (HPET)

6.1 References

None.

6.2 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, and 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.

6.2.1 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 24-MHz crystal as its clock. The accuracy of the main counter is as accurate as the crystal that is used in the system.

6.2.2 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 24-MHz 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 24-MHz clock during boot time to an accuracy that ensures the error introduced by this off-load is less than 10 ppb (.000001%).

When the 24-MHz clock is active, the 64-bit counter will increment by one each cycle of the 24-MHz clock when enabled. When the 24-MHz 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.

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

The HPET timer in the PCH runs typically on the 24-MHz 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 24-MHz clock to shut down when it has no active comparators.

6.2.3 Off-loadable Timer

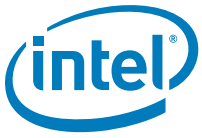
The Off-loadable Timer Block consists of a 64b fast clock counter and an 82b slow clock counter. During fast clock mode the counter increments by one on every rising edge of the fast clock. During slow clock mode, the 82-bit slow clock counter will increment by the value provided by the Off-load Calibrator.

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 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 clocks 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.



6.2.4 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.

6.2.4.1 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.			

6.2.4.2 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 using processor message interrupts.

6.2.4.3 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 processor message interrupt delivery option has HIGHER priority and is mutually exclusive to the standard interrupt delivery option. Thus, if the Tn_PROCMSG_EN_CNF

bit is set, the interrupts will be delivered directly to the processor, rather than by means of the APIC or 8259.

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

6.2.5 Periodic Versus Non-Periodic Modes

6.2.5.1 Non-Periodic Mode

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.

6.2.5.2 Periodic Mode

Timer 0 is the only timer that supports periodic mode. Refer to Section 2.3.9.2.2 of the *IA-PC HPET Specification* for more details of this mode.

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:

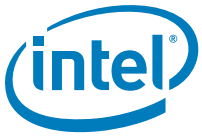
1. Software clears the ENABLE_CNF bit to prevent any interrupts.
2. Software Clears the main counter by writing a value of 00h to it.
3. Software sets the TIMER0_VAL_SET_CNF bit.
4. Software writes the new value in the TIMER0_COMPARATOR_VAL register.
5. Software sets the ENABLE_CNF bit to enable interrupts.

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:

1. Set TIMER0_VAL_SET_CNF bit.
2. Set the lower 32 bits of the Timer0 Comparator Value register.
3. Set TIMER0_VAL_SET_CNF bit.
4. Set the upper 32 bits of the Timer0 Comparator Value register.

6.2.6 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.

6.2.7 Interrupt Levels

Interrupts directed to the internal 8259s are active high. See Section 21.8, "Advanced Programmable Interrupt Controller (APIC) (D31:F0)" 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.

6.2.8 Handling Interrupts

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

6.2.9 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. Provide the PCH temperature in units of 1/2 °C to the EC.
2. Allow programmed trip points to cause alerts via an interrupt (SCI, SMI, and INTx) or shut down the system (unconditionally transitions the system to S5) with a programmable catastrophic trip point.

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.

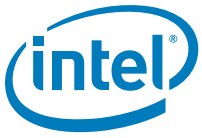
7.1.3 Thermal Sensor Accuracy ($T_{accuracy}$)

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 (0xFF) 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.

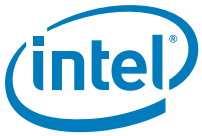
§ §

8 Power and Ground Signals

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

Note: The historical Core well (on in S0 only) and ASW well (on in S0/M0 and Sx/M3) is no longer needed on the PCH due to several new internal power management capabilities. The new Primary well is equivalent to the historical Suspend well such that the supply is on in S0, S3, S4, S5. Refer to the Power Management Chapter for more details.

Name	Description
VCCPRIM_CORE	**Core Logic Primary Well:** This rail operates at 1.0 V. When SLP_S0# is asserted, the rail may optionally be lowered to 0.7 V.
VCCPRIM_1p0	**Primary Well 1.0V:** For I/O blocks, ungated ISH SRAM power, USB AFE Digital Logic, JTAG, Thermal Sensor and MIPI DPHY.
VCCSRAM_1p0	**SRAM Primary Well 1.0V:** Dedicated SRAM rail.
VCCMPHYAON_1p0	**Mod PHY Always On Primary 1.0V**: Always on primary supply for PCIe/DMI/ USB3/SATA/MIPI MPHY logic
VCCMPHYGT_1p0	**Mod PHY Primary 1.0V**: Primary supply for PCIe/DMI/USB3/SATA/MIPI MPHY logic. Note: This supply cannot support external power gating.
VCCAMPHYPLL_1p0	**Analog supply for USB3, PCIe Gen 2/Gen 3, SATA3 and MIPI PLL 1.0V:** Filtering is required.
VCCAPLLEBB_1p0	**PCIe* PLL EBB Primary 1.0:** EBB contains primary supply for PCIe PLL dividers and lane drivers.
VCCAPLL_1p0	**Analog supply for OPI, USB2 and Audio PLL Primary 1.0V:** Filtering is required.
VCCCLK1, VCCCLK2, VCCCLK3, VCCCLK4, VCCCLK5, VCCCLK6	**Clock Buffers Primary 1.0V:** Filtering is required.
VCCPGPPA	**Group A Primary Well GPIOs 3.3V or 1.8V**
VCCPGPPB	**Group B Primary Well GPIOs 3.3V or 1.8V**
VCCPGPPC	**Group C Primary Well GPIOs 3.3V or 1.8V**
VCCPGPPD	**Group D Primary Well GPIOs 3.3V or 1.8V**
VCCPGPPE	**Group E Primary Well GPIOs 3.3V or 1.8V**
VCCPGPPF	**Group F Primary Well GPIOs 1.8V only**
VCCPGPPG	**Group G Primary Well GPIOs 3.3V or 1.8V**
VCCATS	**Thermal Sensor Primary Well 1.8V**
VCCHDA	**HD Audio Power 3.3V, 1.8V, 1.5V.** For Intel High Definition Audio.
VCCSPI	**SPI Primary Well 3.3V or 1.8V**
VCCPRIM_3p3	**Primary Well 3.3V**
VCCRTCPRIM_3p3	**RTC Logic Primary Well 3.3V.** This power supplies the RTC internal VRM. It will be off during Deep Sx mode.
DCPDSW_1p0	**Deep Sx Well 1.0V:** This rail is generated by on die DSW low dropout (LDO) linear voltage regulator to supply DSW GPIOs, DSW core logic, and DSW USB2 logic. Board needs to connect 1 uF capacitor to this rail and power should NOT be driven from the board. When primary well power is up, this rail is bypassed from VCCPRIM_1p0.
VCCDSW_3p3	**Deep Sx Well for GPD GPIOs and USB2**
DCPRTC	**RTC de-coupling capacitor only.** This rail should NOT be driven.



Name	Description
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. This is due to timing issue that may cause Data/RTC corruption. Refer to the Platform Design Guide, RTC Design Guidelines chapter and doc#549657 Design Considerations for Platforms Without a Coin Cell Battery white papers ***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.
VSS	**Ground**

§ §

9 Pin Straps

The following signals are used for static configuration. They are sampled at the rising edge of RSMRST# or PCH_PWROK to select configuration and then revert later to their normal usage. To invoke the associated mode, the signal should be driven at least four PCI clocks prior to the time it is sampled.

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 Management Engine and the Host system.

Table 9-1. Functional Strap Definitions (Sheet 1 of 3)

Signal	Usage	When Sampled	Comment
SPKR/GPP_B14	Top Swap Override	Rising edge of PCH_PWROK	The signal has a weak 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 PLTRST# de-asserts. 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.
GSPI0_MOSI/ GPP_B18	No Reboot	Rising edge of PCH_PWROK	The signal has a weak 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 PLTRST# de-asserts. 2. This signal is in the primary well.
SMBALERT#/ GPP_C2	TLS Confi-dentiality	Rising edge of RSMRST#	This signal has a weak internal Pull-down. 0 = **Disable** Intel ME Crypto Transport Layer Security (TLS) cipher suite (no confidentiality). (Default) 1 = **Enable** Intel ME Crypto Transport Layer Security (TLS) cipher suite (with confidentiality). Must be pulled up to support Intel AMT with TLS and Intel SBA (Small Business Advantage) with TLS. ***Notes:*** 1. The internal Pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.

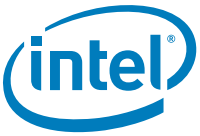
Table 9-1. Functional Strap Definitions (Sheet 2 of 3)

Signal	Usage	When Sampled	Comment
GSPI1_MOSI/ GPP_B22	Boot BIOS Strap Bit BBS	Rising edge of PCH_PWROK	This Signal has a weak internal Pull-down. This field determines the destination of accesses to the BIOS memory range. Also controllable using Boot BIOS Destination bit (Bus0, Device31, Function0, offset BCh, bit 6). **Bit 6** — **Boot BIOS Destination** 0 — SPI (Default) 1 — LPC ***Notes:*** 1. The internal Pull-down is disabled after PLTRST# de-asserts. 2. If option 1 (LPC) is selected, BIOS may still be placed on LPC, but all platforms are required to have SPI flash connected directly to the PCH's SPI bus with a valid descriptor in order to boot. 3. Boot BIOS Destination select to LPC by functional strap or using Boot BIOS Destination bit will not affect SPI accesses initiated by Intel ME or Integrated GbE LAN. 4. This signal is in the primary well.
SML0ALERT#/ GPP_C5	eSPI or LPC	Rising edge of RSMRST#	This signal has a weak internal Pull-down. 0 = **LPC** Is selected for EC. (Default) 1 = **eSPI** Is selected for EC. ***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#	This signal has an internal Pull-up. This strap should sample HIGH. There should NOT be any onboard devices driving it to opposite direction during strap sampling.
SPI0_MISO	Reserved	Rising edge of RSMRST#	This signal has an internal Pull-up. This strap should sample HIGH. There should NOT be any on-board devices driving it to opposite direction during strap sampling.
SML1ALERT#/ PCHHOT#/ GPP_B23	Reserved	Rising edge of RSMRST#	This signal has an internal Pull-down. This strap should sample LOW. There should NOT be any on-board devices driving it to opposite direction during strap sampling. ***Note:*** When used as PCHHOT#, a 150k weak board Pull-up is recommended to ensure it does not override the internal Pull-down strap sampling.
SPI0_IO2	Reserved	Rising edge of RSMRST#	This signal has an internal Pull-up. This strap should sample HIGH. There should NOT be any on-board devices driving it to opposite direction during strap sampling.
SPI0_IO3	Reserved	Rising edge of RSMRST#	This signal has an internal Pull-up. This strap should sample HIGH. There should NOT be any on-board devices driving it to opposite direction during strap sampling.

Table 9-1. Functional Strap Definitions (Sheet 3 of 3)

Signal	Usage	When Sampled	Comment
HDA_SDO / I2S_TXD0	Flash Descriptor Security Override	Rising edge of PCH_PWROK	This signal has a weak 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 PLTRST# de-asserts. 2. Asserting HDA_SDO high on the rising edge of PCH_PWROK will also halt Intel Management Engine after Chipset bring up and disable runtime Intel ME features. This is a debug mode and must not be asserted after manufacturing/debug. 3. This signal is in the primary well.
DDPB_CTRLDATA/ GPP_E19	Display Port B Detected	Rising edge of PCH_PWROK	This signal has a weak internal Pull-down. 0 = Port B is not detected. (Default) 1 = Port B is detected. ***Notes:*** 1. The internal Pull-down is disabled after PLTRST# de-asserts. 2. This signal is in the primary well.
DDPC_CTRLDATA/ GPP_E21	Display Port C Detected	Rising edge of PCH_PWROK	This signal has a weak internal Pull-down. 0 = Port C is not detected. (Default) 1 = Port C is detected. ***Notes:*** 1. The internal Pull-down is disabled after PLTRST# de-asserts. 2. This signal is in the primary well.

§ §



10 Electrical Characteristics

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

10.1 Absolute Maximum Ratings

Table 10-1. PCH Absolute Maximum Ratings

Parameter	Maximum Limits
Voltage on any 0.95V Pin with respect to Ground	-0.5 to 1.04V
Voltage on any 1.0V Pin with respect to Ground	-0.5 to 1.3V
Voltage on any 1.5V Pin with respect to Ground	-0.5 to 2.0V
Voltage on any 1.8V Pin with respect to Ground	-0.5 to 2.3V
Voltage on any 3.3V Pin with respect to Ground	-0.7 to 3.7V

Table 10-1 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 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 PCH Power Supply Range

Table 10-2. PCH Power Supply Range

Power Supply	Minimum	Maximum
0.95V	0.90V	1.00V
1.00V	0.95V	1.05V
1.50V	1.43V	1.58V
1.80V	1.71V	1.89V
3.30V	3.13V	3.46V

10.3 General DC Characteristics

Table 10-3. PCH-U Measured I_{cc}[4] (Sheet 1 of 2)

Voltage Rail	Voltage (V)	S0 Iccmax Current[3] (A)	Sx Icc Idle Current[5] (mA)	Deep Sx Icc Idle Current (mA)	G3 (µA)
VCCMPHYPLL_1p0	1.0	0.088	0.350	0	0
VCCAPLLEBB_1p0	1.0	0.033	1.00	0	0
VCCAPLL_1p0	1.0	0.026	0.350	0	0
VCCATS	1.8	0.006	0.100	0	0
VCCCLK1	1.0	0.035	0.014	0	0
VCCCLK2	1.0	0.029	0.015	0	0
VCCCLK3	1.0	0.024	0.011	0	0
VCCCLK4	1.0	0.033	0.010	0	0
VCCCLK5	1.0	0.004	1.00	0	0
VCCCLK6	1.0	0.039	0.350	0	0
VCCMPHYAON_1p0	1.0	0.022	3.00	0	0
VCCMPHYGT_1p0	1.0	See Table 10-5	7.40	0	0
VCCPGPPA	3.3	0.020	0.350	0	0
	1.8	0.009	0.350	0	0
VCCPGPPB	3.3	0.004	0.350	0	0
	1.8	0.002	0.350	0	0
VCCPGPPC	3.3	0.006	0.350	0	0
	1.8	0.003	0.350	0	0
VCCPGPPD	3.3	0.008	0.350	0	0
	1.8	0.003	0.350	0	0
VCCPGPPE	3.3	0.006	0.350	0	0
	1.8	0.002	0.350	0	0
VCCPGPPF	1.8	0.161	0.350	0	0
VCCPGPPG	3.3	0.041	0.350	0	0
	1.8	0.056	0.350	0	0
VCCHDA	3.3	0.068	0.350	0	0
	1.8	0.040	0.350	0	0
	1.5	0.030	0.350	0	0
VCCPRIM_CORE	VCC CORE	2.574	40.0	0	0
VCCPRIM_1p0	1.0	0.696	8.65	0	0
VCCPRIM_3p3	3.3	0.075	1.00	0	0
VCCSPI	3.3	0.011	0.350	0	0
	1.8	0.007	0.350	0	0
VCCSRAM_1p0	1.0	0.642	3.61	0	0
VCCDSW_3p3	3.3	0.118	2.19	2.19	0
VCCRTCPRIM_3p3	3.3	0.200 mA	0.100	0	0

Table 10-3. PCH-U Measured I_{CC}[4] (Sheet 2 of 2)

Voltage Rail	Voltage (V)	S0 Iccmax Current[3] (A)	Sx Icc Idle Current[5] (mA)	Deep Sx Icc Idle Current (mA)	G3 (µA)
VCCRTC	3.0	0.200 mA	0.100	0.100	6 Notes 1, 2

Notes:
1. G3 state shown to provide an estimate of battery life.
2. Icc (RTC) data is taken with VCCRTC at 3.0V while the system is in a mechanical off (G3) state at room temperature.
3. Iccmax estimates assume 110 °C.
4. The Iccmax value is a steady state current that can happen after respective power ok has asserted (or reset signal has de-asserted).
5. Sx Icc Idle assumes PCH is idle and ME is power gated.

Table 10-4. PCH-Y Measured I_{CC}[4] (Sheet 1 of 2)

Voltage Rail	Voltage (V)	S0 Iccmax Current[3] (A)	Sx Icc Idle Current[5] (mA)	Deep Sx Icc Idle Current (mA)	G3 (µA)
VCCMPHYPLL_1p0	1.0	0.088	0.350	0	0
VCCAPLLEBB_1p0	1.0	0.033	1.0	0	0
VCCAPLL_1p0	1.0	0.026	0.350	0	0
VCCATS	1.8	0.006	0.100	0	0
VCCCLK1	1.0	0.035	0.350	0	0
VCCCLK2	1.0	0.029	0.015	0	0
VCCCLK3	1.0	0.024	0.011	0	0
VCCCLK4	1.0	0.033	0.010	0	0
VCCCLK5	1.0	0.004	1.00	0	0
VCCCLK6	1.0	0.039	0.350	0	0
VCCMPHYAON_1p0	1.0	0.022	3.0	0	0
VCCMPHYGT_1p0	1.0	See Table 10-5	7.4	0	0
VCCPGPPA	3.3	0.020	0.350	0	0
	1.8	0.009	0.350	0	0
VCCPGPPB	3.3	0.004	0.350	0	0
	1.8	0.002	0.350	0	0
VCCPGPPC	3.3	0.006	0.350	0	0
	1.8	0.003	0.350	0	0
VCCPGPPD	3.3	0.008	0.350	0	0
	1.8	0.003	0.350	0	0
VCCPGPPE	3.3	0.006	0.350	0	0
	1.8	0.002	0.350	0	0
VCCPGPPF	1.8	0.161	0.350	0	0
VCCPGPPG	3.3	0.041	0.350	0	0
	1.8	0.056	0.350	0	0

Table 10-4. PCH-Y Measured I_{cc}[4] (Sheet 2 of 2)

Voltage Rail	Voltage (V)	S0 Iccmax Current[3] (A)	Sx Icc Idle Current[5] (mA)	Deep Sx Icc Idle Current (mA)	G3 (µA)
VCCHDA	3.3	0.068	0.350	0	0
	1.8	0.036	0.350	0	0
	1.5	0.033	0.350	0	0
VCCPRIM_CORE	VCC CORE	1.1	40.0	0	0
VCCPRIM_1p0	1.0	0.599	9.28	0	0
VCCPRIM_3p3	3.3	0.075	0.351	0	0
VCCSPI	3.3	0.011	0.350	0	0
	1.8	0.007	0.350	0	0
VCCSRAM_1p0	1.0	0.565	3.61	0	0
VCCDSW_3p3	3.3	0.071	2.19	2.0	0
VCCRTCPRIM_3p3	3.3	0.200 mA	0.100	0	0
VCCRTC	3.0	0.200 mA	0.100	0.100	6 Notes 1, 2

Notes:
1. G3 state shown to provide an estimate of battery life.
2. Icc (RTC) data is taken with VCCRTC at 3.0V while the system is in a mechanical off (G3) state at room temperature.
3. Iccmax estimates assume 110 °C.
4. The Iccmax value is a steady state current that can happen after respective power ok has asserted (or reset signal has de-asserted).
5. Sx Icc Idle assumes PCH is idle and ME is power gated.

Table 10-5. PCH-U/Y VCCMPHY_1p0 Icc Adder Per HSIO Lane

Icc (A)	Details
0.064	All HSIO disabled
0.154	Each PCIe Gen3 Lane
0.102	Each PCIe Gen2 Lane
0.132	Each USB3 Port
0.099	SSIC
0.044	GbE Port
0.132	Each SATA Gen3 Port

Table 10-6. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 1 of 5)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
Associated Signals[3]: GPP_A[0-23], ESPI, LAD, ISH, GPP_B[0-22], GPP_C[8:15], GPP_C[20:23], GPP_D[0:4], GPP_D[9:12], GPP_D[15:16], GPP_E[13:17], GPP_F[0:3], GPD[0:11], SPI0_IO2, SPI0_IO3, HDA_SDO/I2S0_TXD, HDA_SYNC/I2S0_SFRM, HDA_SDI1/I2S1_RXD, HDA_RST#/ I2S1_SCLK, I2S1_SFRM, HDA_BLK/I2S0_SCLK, HDA_SDI0/ I2S0_RXD, I2S1_TXD, SPI0_MOSI, SPI0_MISO, SPI0_CS2#, SPI0_CS0#, SPI0_CS1#, SPI0_CLK, CL_RST#, SYS_PWROK, SYS_RESET#, DDPB_CTRLCLK/GPP_E18, DDPB_CTRLDATA/GPP_E19, DDPC_CTRLCLK/GPP_E20, DDPC_CTRLDATA/GPP_E21							
3.3V Operation							
Input	V_{IH}	Input High Voltage	0.65 x V_{CC}	V_{CC} + 0.4	V		1
	V_{IL}	Input Low Voltage	-0.5	0.35 x V_{CC}	V		2
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance		3	pF		
Output	V_{OH}	Output High Voltage	0.9 * V_{CC}	V_{CC}	V	I_{oh}=0.5mA	4
	V_{OL}	Output Low Voltage	0	0.4	V	I_{ol}=-4mA	4
	R_{pu}	WPU Resistance	5K-30% 20K-30%	5K+30% 20K+30%			
	R_{pd}	WPD Resistance	5K-30% 20K-30%	5K+30% 20K+30%			
1.8V Operation							
Input	V_{IH}	Input High Voltage	0.65 x V_{CC}	V_{CC} + 0.4	V		
	V_{IL}	Input Low Voltage	-0.5	0.35 x V_{CC}	V		
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance		3	pF		
Output	V_{OL}	Output Low Voltage		0.4	V	I_{ol}=-4mA	4
	R_{pu}	WPU Resistance	5K-30% 20K-30%	5K+30% 20K+30%			
	R_{pd}	WPD Resistance	5K-30% 20K-30%	5K+30% 20K+30%			

Notes:
1. V_{IH} for LPC=0.5*VCC and VIH for HD Audio =0.6*VCC (*1.5V supply operation).
2. V_{IL} for LPC=0.3*VCC and VIH for HD Audio =0.4*VCC (*1.5V supply operation).
3. For GPIO supported voltages, refer to the GPIO chapter.
4. Each GPIO pin can support 3mA Ioh/Iol Max.

Table 10-6. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 2 of 5)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
Associated Signals[1]: GPP_B23, GPP_C[0:7], GPP_C[16:19], GPP_D[5:8], GPP_D[13:14], GPP_E[18:23], GPP_F[4:11], eDP_BKLCTL, eDP_BKLEN, eDP_VDDEN							
3.3V Operation							
Input	V_{IH}	Input High Voltage	0.65 x V_{CC}	V_{CC} + 0.4	V		
	V_{IL}	Input Low Voltage	-0.5	0.35 x V_{CC}	V		
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance		3.5	pF		
Output	V_{OH}	Output High Voltage	V_{CC} - 0.5	V_{CC}	V	I_{oh}=1mA	
	V_{OL}	Output Low Voltage		0.4	V	I_{ol}=-4mA	2
	R_{pu}	WPU Resistance	5K-30% 20K-30%	5K+30% 20K+30%			
	R_{pd}	WPD Resistance	5K-30% 20K-30%	5K+30% 20K+30%			
1.8V Operation							
Input	V_{IH}	Input High Voltage	0.65 x V_{CC}	V_{CC} + 0.4	V		
	V_{IL}	Input Low Voltage	-0.5	0.35 x V_{CC}	V		
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance		3.5	pF		
Output	V_{OH}	Output High Voltage	V_{CC} - 0.5	V_{CC}	V	I_{oh}=1mA	
	V_{OL}	Output Low Voltage		0.4	V	I_{ol}=-4mA	2
	R_{pu}	WPU Resistance	5K-30% 20K-30%	5K+30% 20K+30%	KΩ	V_{pad}=V_{CC}/2	
	R_{pd}	WPD Resistance	5K-30% 20K-30%	5K+30% 20K+30%	KΩ	V_{pad}=V_{CC}/2	

Notes:
1. For GPIO supported voltages, refer to GPIO chapter.
2. Each GPIO pin can support 3mA Ioh/Iol Max.

Table 10-6. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 3 of 5)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
Associated Signals: PECI, PROCPWRGD, THERMTRIP#, PCH_JTAG_TDO, JTAGX, PCH_TRST, PCH_JTAG_TDI, PCH_JTAG_TMS, PCH_JTAG_TCK, ITP_PMODE							
Input	V_{IH}	Input High Voltage	PECI: 0.725 * V_{CC} JTAG: 0.8 * V_{CC} CMOS: 0.7 * V_{CC} iDISPLAY: 0.65 * V_{CC}	V_{CC} + 0.25	V		
	V_{IL}	Input Low Voltage	-0.5	PECI: 0.275 * V_{CC} JTAG: 0.51 * V_{CC} CMOS/iDISPLAY: 0.3 * V_{CC}	V		
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance	—	2	pF		
Output	V_{OH}	Output High Voltage	PECI: 0.75 * VCC	V_{CC}	V	I_{oh}=-6mA	
	V_{OL}	Output Low Voltage	0	PECI: 0.25 * V_{CC}	V	I_{ol}=0.5mA	
	R_{pu}	WPU Resistance	1K-30% 20K-30%	1K+30% 20K+30%			
	R_{pd}	WPD Resistance	1K-30% 20K-30%	1K+30% 20K+30%			
Associated Signals[1]: GPP_D17/DMIC_CLK1, GPP_D18/DMIC_DATA1, GPP_D19/DMIC_CLK0, GPP_D20/DMIC_DATA0, GPP_D21, GPP_D22, GPP_D23/I2S_SCLK, GPP_E0/SATAXPCIE0/SATAGP0, GPP_E1/SATAXPCIE1/SATAGP1, GPP_E2/SATAXPCIE2/SATAGP2, GPP_E3/CPU_GP0, GPP_E4/ DEVSLP0, GPP_E5/DEVSLP1, GPP_E6/DEVSLP2, GPP_E7/CPU_GP1, GPP_E8/SATALED#, GPP_E9/ USB2_OC0#, GPP_E10/USB2_OC1#, GPP_E11/USB2_OC2#, GPP_E12/USB2_OC3#.							
3.3V Operation							
Input	V_{IH}	Input High Voltage	0.65 x V_{CC}	V_{CC} + 0.4	V		
	V_{IL}	Input Low Voltage	-0.5	0.35 x V_{CC}	V		
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance		3	pF		
Output	V_{OH}	Output High Voltage	0.9 x V_{CC}	V_{CC}	V	I_{oh}=0.5mA	2
	V_{OL}	Output Low Voltage		0.1 x VCC	V	I_{ol}=-1.5mA	2
	R_{pu}	WPU Resistance	5K-30% 20K-30%	5K+30% 20K+30%			
	R_{pd}	WPD Resistance	5K-30% 20K-30%	5K+30% 20K+30%			

Table 10-6. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 4 of 5)

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
1.8V Operation							
Input	V_{IH}	Input High Voltage	0.65 x V_{CC}	V_{CC}	V		
	V_{IL}	Input Low Voltage	-0.5	0.35 x V_{CC}	V		
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance		3	pF		
Output	V_{OL}	Output Low Voltage		0.4	V	I_{ol}=-4mA	2
	R_{pu}	WPU Resistance	5K-30% 20K-30%	5K+30% 20K+30%	KΩ	V_{pad}=V_{CC}/2	
	R_{pd}	WPD Resistance	5K-30% 20K-30%	5K+30% 20K+30%	KΩ	V_{pad}=V_{CC}/2	
Notes: 1. For GPIO supported voltages, refer to GPIO chapter. 2. Each GPIO pin can support 3mA Ioh/Iol Max.							
Associated Signals[1]:GPP_F12/EMMC_CMD, GPP_F13/EMMC_DATA0, GPP_F14/EMMC_DATA1, GPP_F15/EMMC_DATA2, GPP_F16/EMMC_DATA3, GPP_F17/EMMC_DATA4, GPP_F18/EMMC_DATA5, GPP_F19/EMMC_DATA6, GPP_F20/EMMC_DATA7, GPP_F21/EMMC_RCLK, GPP_F22/EMMC_CLK, GPP_F23, GPP_G0/SD_CMD, GPP_G1/SD_DATA0, GPP_G2/SD_DATA1, GPP_G3/SD_DATA2, GPP_G4/SD_DATA3, GPP_G5/SD_CD#, GPP_G6/SD_CLK, GPP_G7/SD_WP.							
3.3V Operation							
Input	V_{IH}	Input High Voltage	0.625 x V_{CC}	V_{CC} + 0.4	V		
	V_{IL}	Input Low Voltage		0.25 x V_{CC}	V		
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance		3	pF		
Output	V_{OH}	Output High Voltage	0.75 x V_{CC}		V	I_{oh}=0.1mA	2
	V_{OL}	Output Low Voltage	0	0.125 x V_{CC}	V	I_{ol}=0.1mA	2
	R_{pu}	WPU Resistance	5K-30% 20K-30%	5K+30% 20K+30%			
	R_{pd}	WPD Resistance	5K-30% 20K-30%	5K+30% 20K+30%			
1.8V Operation							
Input	V_{IH}	Input High Voltage	eMMC: 0.65 * V_{CC}	V_{CC} + 0.4	V		
	V_{IL}	Input Low Voltage		eMMC: 0.35 * V_{CC}	V		
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance		3	pF		

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

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
Output	V_{OH}	Output High Voltage	eMMC: V_{CC} - 0.45		V	I_{oh}=2mA	2
	V_{OL}	Output Low Voltage		0.45	V	I_{ol}=2mA	2
	R_{pu}	WPU Resistance	5K-30% 20K-30%	5K+30% 20K+30%			
	R_{pd}	WPD Resistance	5K-30% 20K-30%	5K+30% 20K+30%			
Notes: 1. For GPIO supported voltages, refer to GPIO chapter. 2. Each GPIO pin can support 3mA Ioh/Iol Max.							
Associated Signals: CL_DATA, CL_CLK.							
	CL_V_{Ref}	Supply Voltage Reference	0.392	0.408	V		
Input	V_{IH}	Input High Voltage	CL_V_{ref} + 0.075		V		
	V_{IL}	Input Low Voltage		CL_V_{ref} - 0.075	V		
	I_{IL}	Input Leakage Current	-10	10	µA		
	C_{IN}	Input Pin Capacitance		2	pF		
Output	V_{OH}	Output High Voltage	0.61	0.98	V	R_{load} = 100 ohm to GND	1
	V_{OL}	Output Low Voltage	0	0.15	V	I_{ol}=1mA	
	R_{pu}	WPU Resistance	20K-30%	20K+30%			
	R_{pd}	WPD Resistance	20K-30%	20K+30%			
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.							

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	0.65 x V_{CCRTC}	V_{CCRTC}+0.5	V		4, 6
	V_{IL}	Input Low Voltage	-0.5	0.3 x V_{CCRTC}	V		6
Associated Signals: RTCRST#							
Input	V_{IH}	Input High Voltage	0.75 x V_{CCRTC}	V_{CCRTC}+0.5	V		4, 5, 6
	V_{IL}	Input Low Voltage	-0.5	0.4 x V_{CCRTC}	V		6

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

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
Associated Signals: RTCX1#							
Input	V_{IH}	Input High Voltage	0.8	1.2	V		
	V_{IL}	Input Low Voltage	-0.5	0.1	V		
Associated Signals: XTAL24_IN							3
Input	V_{IH}	Input High Voltage	0.8	1.2	V		
	V_{IL}	Input Low Voltage	-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. V_{CCRTC} is the voltage applied to the V_{CCRTC} 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. See 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. Differential Signals Characteristics (Sheet 1 of 4)

Symbol		Parameter	Minimum	Maximum	Unit	Conditions	Notes
Associated Signals: eMMC*							
1.8V							
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	0.1	1.2	V		
VRX_CM-ACp		AC peak Common Mode Input Voltage	—	150	mV		
Associated Signals: MIPI* CSI2: Refer to MIPI® Alliance D-PHY Specification 1.1.							
HS Input	VRX_CM-ACp	Common-mode voltage HS receives mode	70	330	mV		
	VRX-DIFF p-p	Differential input high threshold	70		mV		
	VRX-DIFF L	Differential input low threshold		-70	mV		
	V_{IH}	Single-ended input high voltage		460	mV		
	V_{IL}	Single-ended input low voltage	-40		mV		
	VTerm_Enable	Single-ended threshold for HS termination enable		450	mV		
	ZTX-DIFF-DC	DC Differential RX impedance	80	125	Ohm		

Table 10-8. Differential Signals Characteristics (Sheet 2 of 4)

Symbol		Parameter	Minimum	Maximum	Unit	Conditions	Notes
LP Input	V_{IH}	Logic 1 input voltage	880		mV		
	V_{IL}	Logic 0 input voltage		550	mV		
	V_{HYST}	Input hysteresis	25		mV		
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		
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* GEN3 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

Table 10-8. Differential Signals Characteristics (Sheet 3 of 4)

Symbol	Parameter	Minimum	Maximum	Unit	Conditions	Notes
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
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	Iol=5 mA	6
VOH	Output High Voltage	3.3V – 0.5	—	V	Ioh=-2mA	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
Note:	VDI VCM, VSE, VCRS, VOL, VOH are USB 2.0 FS/LS electrical characteristic.					
Associated Signals: USB 3.0						
VTX-DIFF-PP	Differential Peak to Peak Output Voltage	0.8	1.2	V		

Table 10-8. Differential Signals Characteristics (Sheet 4 of 4)

Symbol		Parameter	Minimum	Maximum	Unit	Conditions	Notes
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		
Associated Signals: RTCX1							
Input	V_{IH}	Input High Voltage	0.8	1.2	V		
	V_{IL}	Input Low Voltage	-0.5	0.1	V		

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 3.0 mVdiff p-p = 2*|USB3Rp[x] – USB3Rn[x]|; USB 3.0 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. V_{Cross} 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 max. voltage including overshoot
16. The min. voltage including undershoot
17. The total variation of all V_{Cross} measurements in any particular system. Note that this is a subset of V_{Cross} MIN/MAX (V_{Cross} absolute) allowed. The intent is to limit V_{Cross} induced modulation by setting V_{Cross_Delta} to be smaller than V_{Cross} absolute.

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

Symbol	Parameter	Min.	Nom.	Max.	Unit	Notes
VCCPRIM_1p0	UngatedSRAM, I/O Blocks, USB AFE, Processor Sideband, JTAG, Thermal Sensor, MIPI* DPHY Primary WellSP	0.950	1.0	1.05	V	1
VCCPRIM_CORE	Core Logic Primary Well	0.950	1.0	1.05	V	1
VCCSRAM	SRAM Primary Well	0.950	1.0	1.05	V	1
VCCCLK1	Clock Buffer 1 Primary Well	0.950	1.0	1.05	V	1
VCCCLK2	Clock Buffer 2 Primary Well	0.950	1.0	1.05	V	1
VCCCLK3	Clock Buffer 3 Primary Well	0.950	1.0	1.05	V	1
VCCCLK4	Clock Buffer 4 Primary Well	0.950	1.0	1.05	V	1
VCCCLK5	Clock Buffer 5 Primary Well	0.950	1.0	1.05	V	1
VCCAPLLEBB_1p0	PCIe PLL EBB Primary Well	0.950	1.0	1.05	V	1
VCCAMPHYPLL_1p0	Analog Supply for USB 3.0, PCIe Gen2, SATA and PCIe* Gen 3 PLL Primary Well	0.950	1.0	1.05	V	1
VCCAMPHYAON_1p0	Mod PHY Always On Primary Well	0.950	1.0	1.05	V	1

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

Symbol	Parameter	Min.	Nom.	Max.	Unit	Notes
VCCMPHYGT_1p0	Mod PHY Externally Gated Primary Well	0.950	1.0	1.05	V	1
VCCAPLL_1p0	Analog Supply for OPI, USB2 and Audio PLL Primary Well	0.950	1.0	1.05	V	1
VCCPGPPA	Group A Primary Well GPIOs	3.13	3.3	3.46	V	1
		1.71	1.8	1.89	V	1
VCCPGPPB	Group B Primary Well GPIOs	3.13	3.3	3.46	V	1
		1.71	1.8	1.89	V	1
VCCPGPPC	Group C Primary Well GPIOs	3.13	3.3	3.46	V	1
		1.71	1.8	1.89	V	1
VCCPGPPD	Group D Primary Well GPIOs	3.13	3.3	3.46	V	1
		1.71	1.8	1.89	V	1
VCCPGPPE	Group E Primary Well GPIOs	3.13	3.3	3.46	V	1
		1.71	1.8	1.89	V	1
VCCPGPPF	Group F Primary Well GPIOs	1.71	1.8	1.89	V	1
VCCPGPPG	Group G Primary Well GPIOs	3.13	3.3	3.46	V	1
		1.71	1.8	1.89	V	1
VCCSPI	SPI Primary Well	3.13	3.3	3.46	V	1
		1.71	1.8	1.89	V	1
VCCATS	Thermal Sensor Primary Well	1.71	1.8	1.89	V	1
VCCHDA	Intel® HD Audio Supply Primary Well	3.13	3.3	3.46	V	1
		1.71	1.8	1.89	V	1
		1.425	1.5	1.575	V	1
VCCPRIM_3p3	Primary Well for HVCMOS and display	3.13	3.3	3.46	V	1
VCCDSW_3p3	Deep Sx Well for GPD and USB 2.0	3.13	3.3	3.46	V	1
VCCRTCPRIM_3p3	RTC Logic Primary Well	3.13	3.3	3.46	V	1
VCCRTC	RTC Well Supply	2.0	3.0	3.2	V	1,2,3

Notes:
1. 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.
2. Maximum Crystal ESR is 50 KOhms.
3. The initial VCCRTC voltage can exceed Vmax of 3.2 V (up to 3.47V) for ~1-week period without concerns about damage to the PCH.

10.4 AC Characteristics

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

Symbol	Parameter	Min.	Max.	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

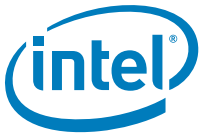


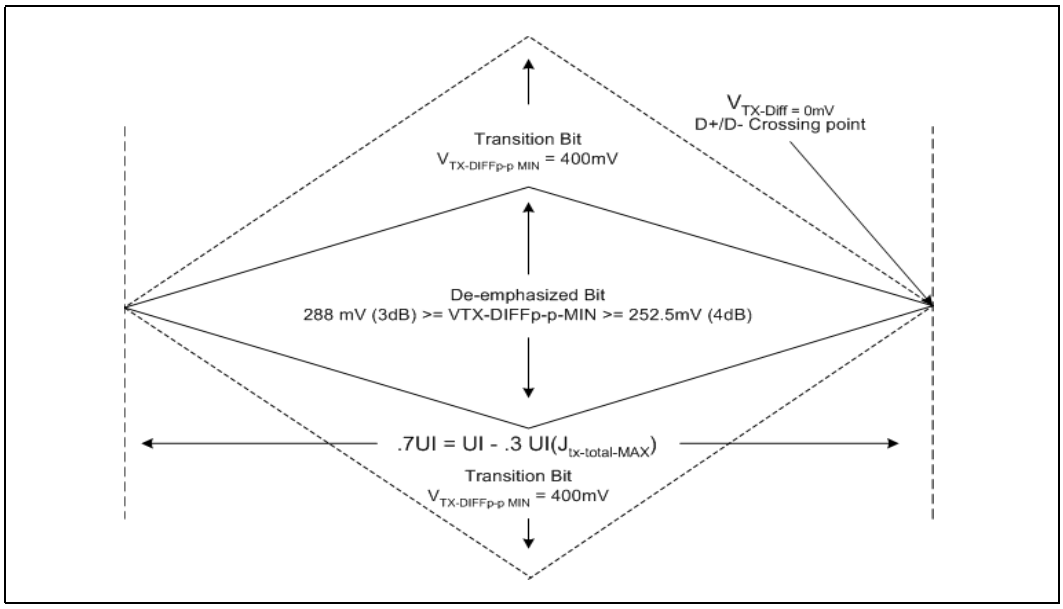
Table 10-10. PCI Express* Interface Timings (Sheet 2 of 2)

Symbol	Parameter	Min.	Max.	Unit	Figures	Notes
Transmitter and Receiver Timings						
UI (GEN3)	Unit Interval – PCI Express*	124.96	125.03	ps		
T_{TX-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		
T_{RX-EYE} (Gen 1)	Minimum Receiver Eye Width	0.40	—	UI	10-2	3,4
T_{RX-EYE} (Gen 2)	Minimum Receiver Eye Width	0.60	—	UI	10-2	3,4
$T_{Min-Pulse}$ (Gen 2)	Instantaneous Pulse Width	0.9	—	UI		

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

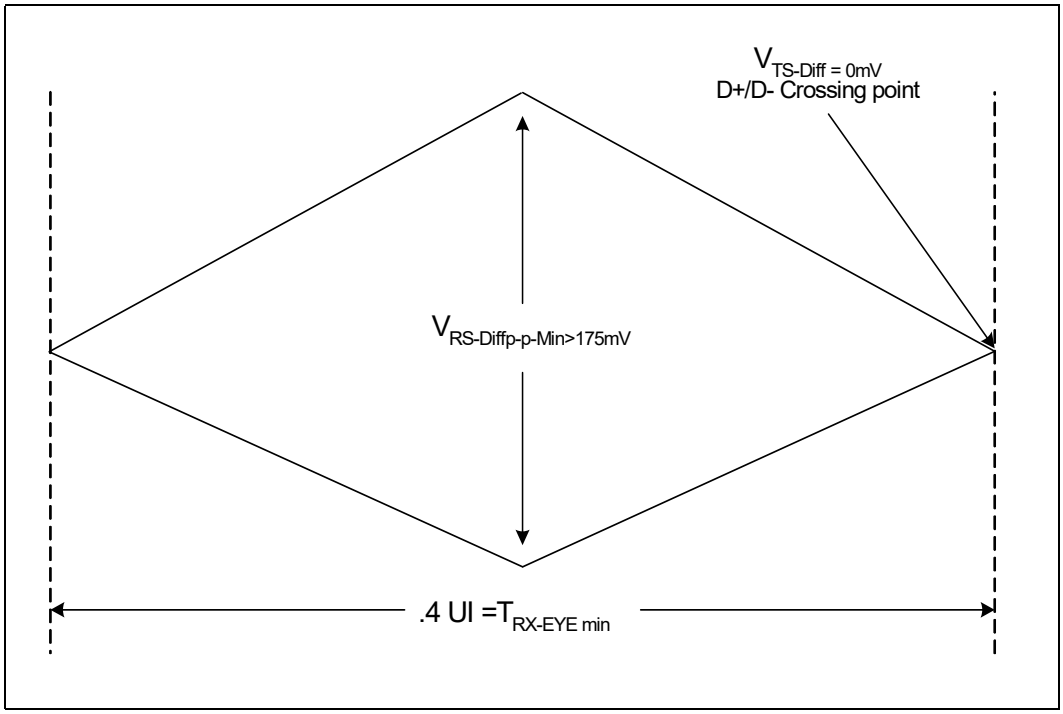
1. Specified at the measurement point into a timing and voltage compliance test load and measured over any 250 consecutive TX UIs. (also refer to the Transmitter compliance eye diagram)
2. A T_{TX-EYE} = 0.70 UI provides for a total sum of deterministic and random jitter budget of $T_{TXJITTER-MAX}$ = 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**
		Max.	**Min.**	**Max.**	**Min.**	**Max.**	
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.

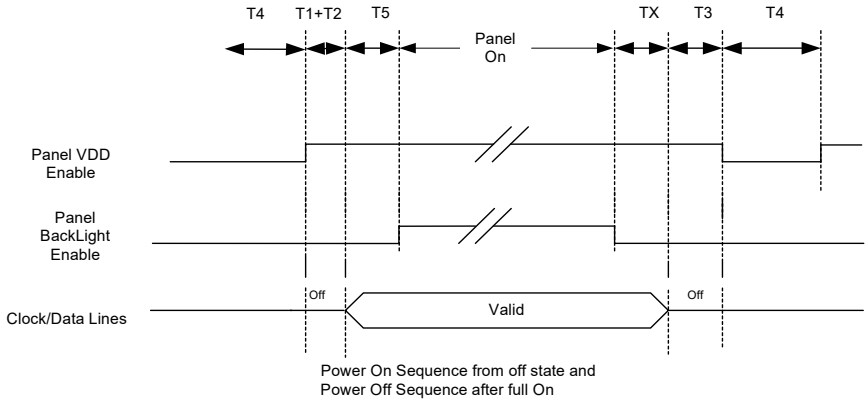
10.4.1 Panel Power Sequencing and Backlight Control

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 parameters TX and T1 through T5 using software is provided.

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

Signal Group: DDPB_HPD0, DDPC_HPD1, DDPD_HPD2, DDPE_HPD3, eDP_HPD						
Symbol	**Parameter**	**Min.**	**Max.**	**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)

Sym	Parameter	Min.	Max.	Unit	Notes	Figure
LPC Clock (CLKOUT_LPC[1:0])						
t1	Period	41.16	42.18	ns		10-4
t2	High Time	16.67	25.00	ns		10-4
t3	Low Time	16.67	25.00	ns		10-4
	Duty Cycle	40	60	%		
	Jitter	—	500	ps	8,9	
	Flight Time (PCH to Device)		3	ns		
CLKOUT_SRC_P/N[5:0] CLKOUT_ITPXDP_[P,N]						
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)						
f_{smb}	Operating Frequency	10	100	KHz		
t18	High Time	4.0	50	µs	2	10-6
t19	Low Time	4.7	—	µs		10-6
t20	Rise Time	—	1000	ns		10-6
t21	Fall Time	—	300	ns		10-6
SMLink[1,0] (SML[1:0]CLK) (Fast Mode: See note 15)						
f_{smb}	Operating Frequency	0	400	KHz		
t18_SMLFM	High Time	0.6	50	µs	2	10-6
t19_SMLFM	Low Time	1.3	—	µs		10-6
t20_SMLFM	Rise Time	—	300	ns		10-6
t21_SMLFM	Fall Time	—	300	ns		10-6
SMLink[1,0] (SML[1,0]CLK) (Fast Mode Plus: See note 17)						
f_{smb}	Operating Frequency	0	1000	KHz		
t18_SMLFMP	High Time	0.26	—	µs	2	10-6

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

Sym	Parameter	Min.	Max.	Unit	Notes	Figure
t19_SMLFMP	Low Time	0.5	—	µs		10-6
t20_SMLFMP	Rise Time	—	120	ns		10-6
t21_SMLFMP	Fall Time	—	120	ns		10-6
HDA_BLK (Intel® High Definition Audio)						
f_{HDA}	Operating Frequency	24.0		MHz		
	Frequency Tolerance	—	100	ppm		
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)						
f_{susclk}	Operating Frequency	32		KHz	4	
t39	High Time	9.5	—	µs	4	
t39a	Low Time	9.5	—	µs	4	
XTAL24_IN/XTAL24_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.) provides 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. A period minimum and maximum include 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 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.

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

Sym	Parameter	Min.	Max.	Unit	Notes	Figure

Notes:
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.

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

Figure 10-4. Clock Timing

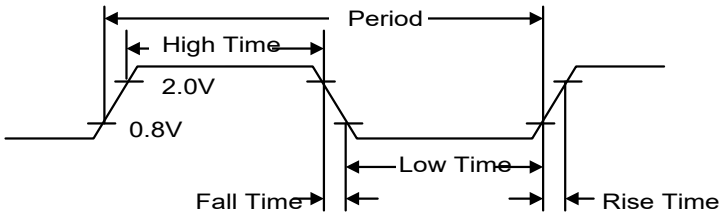


Figure 10-5. Measurement Points for Differential Waveforms

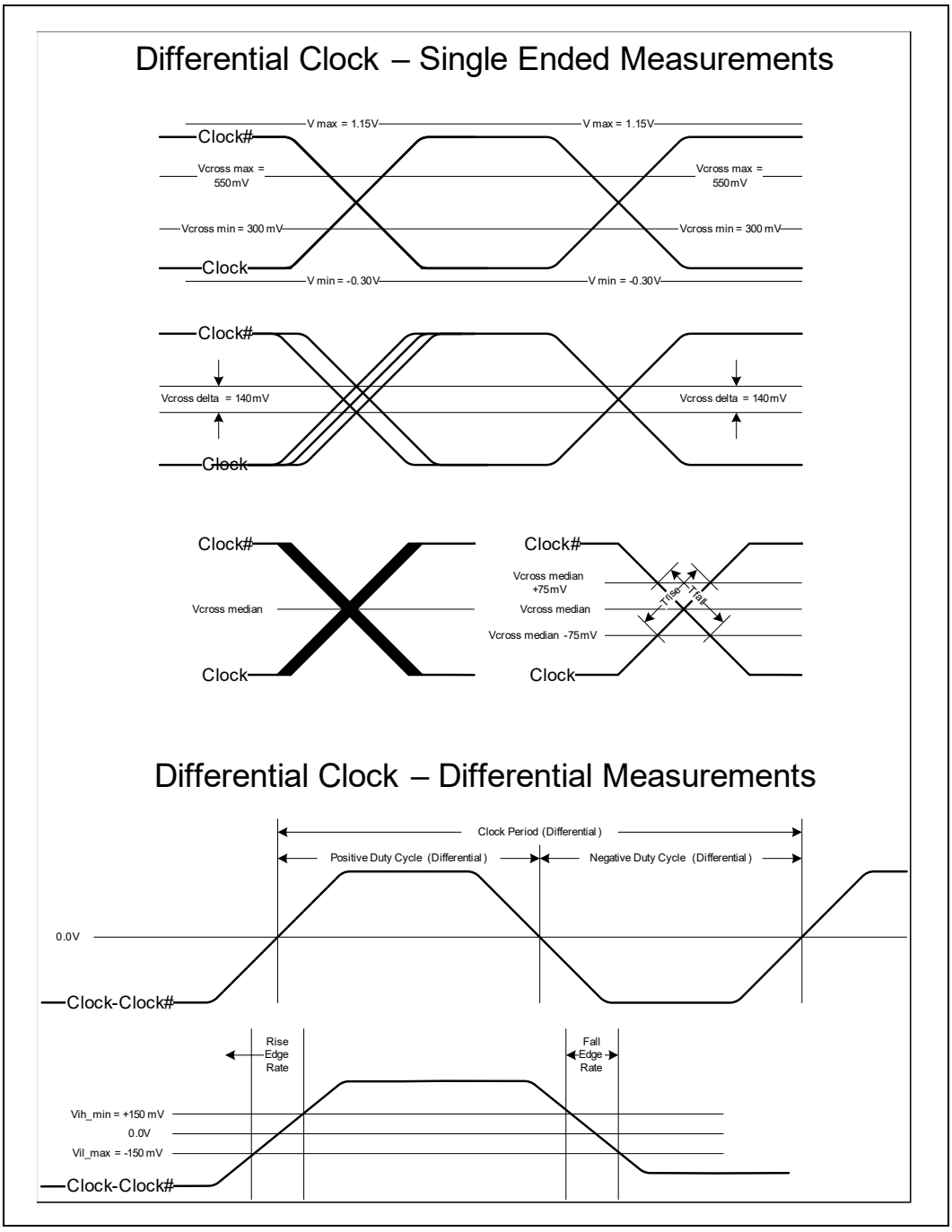
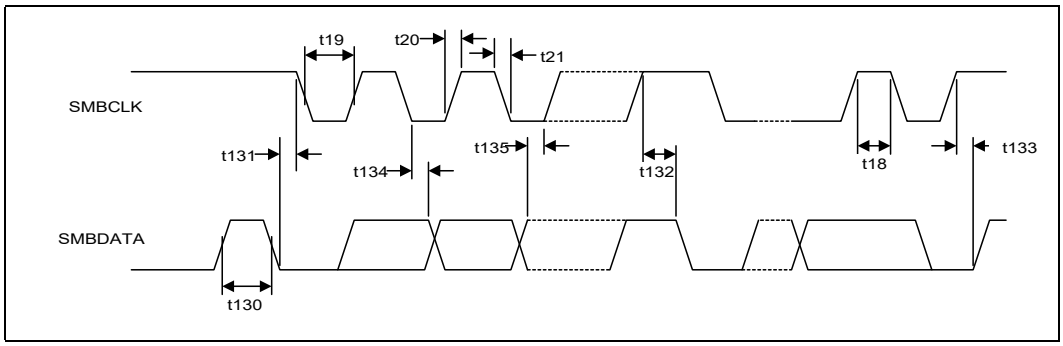


Figure 10-6. SMBus/SMLink Transaction



Note: t*xx* also refers to t*xx*_SM, t*xxx* also refers to t*xxx*SMLFM, SMBCLK also refers to SML[1:0]CLK, and SMBDATA also refers to SML[1:0]DATA.

Figure 10-7. PCH Test Load

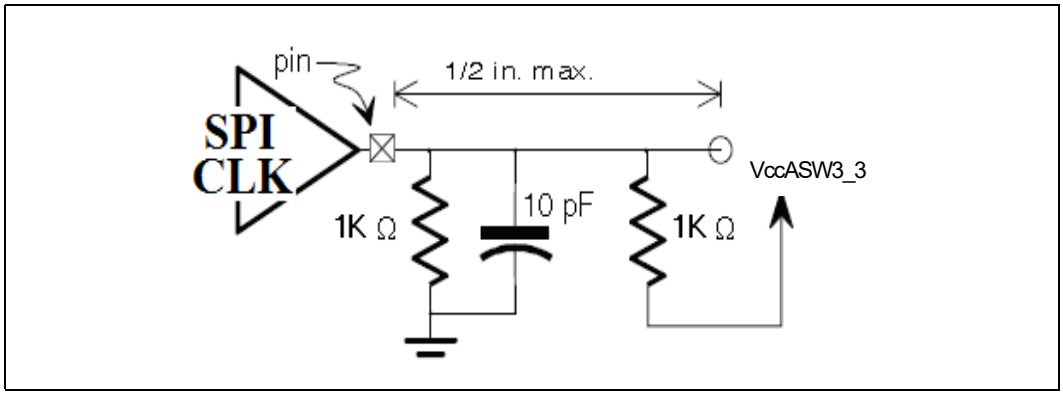


Table 10-14. USB 2.0 Timing (Sheet 1 of 2)

Sym	Parameter	Min.	Max.	Units	Notes	Figure
Full-speed Source (Note 7)						
t100	USBPx+, USBPx- Driver Rise Time	4	20	ns	1,6 C_L = 50 pF	10-8
t101	USBPx+, USBPx- Driver Fall Time	4	20	ns	1,6 C_L = 50 pF	10-8
t102	Source Differential Driver Jitter - To Next Transition - For Paired Transitions	–3.5 –4	3.5 4	ns ns	2, 3	10-9
t103	Source SE0 interval of EOP	160	175	ns	4	10-10
t104	Source Jitter for Differential Transition to SE0 Transition	–2	5	ns	5	
t105	Receiver Data Jitter Tolerance - T o Next Transition - For Paired Transitions	–18.5 –9	18.5 9	ns ns	3	10-9
t106	EOP Width: Receiver must accept EOP	82	—	ns	4	10-10
t107	Width of SE0 interval during differential transition	—	14	ns		

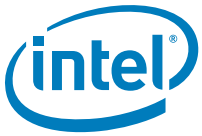


Table 10-14. USB 2.0 Timing (Sheet 2 of 2)

Sym	Parameter	Min.	Max.	Units	Notes	Figure
Low-Speed Source (Note 8)						
t108	USBPx+, USBPx – Driver Rise Time	75	300	ns	1,6 C_L = 200 pF C_L = 600 pF	10-8
t109	USBPx+, USBPx – Driver Fall Time	75	300	ns	1,6 C_L = 200 pF C_L = 600 pF	10-8
t110	Source Differential Driver Jitter To Next Transition For Paired Transitions	–25 –14	25 14	ns ns	2,3	10-9
t111	Source SE0 interval of EOP	1.25	1.50	µs	4	10-10
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	10-9
t114	EOP Width: Receiver must accept EOP	670	—	ns	4	10-10
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 3.0 Interface Transmit and Receiver Timings

Sym	Parameter	Minimum	Maximum	Units	Notes	Figure
UI	Unit Interval – USB 3.0 (5.0 GT/s)	199.9	200.1	ps		
T_{TX-EYE}	Minimum Transmission Eye Width	0.625	—	UI		
P_{U3}	Polling Period U3 State	-	100	mS		
$P_{RX-Detect}$	Polling Period Rx Detect	-	100	mS		

Table 10-16. SSIC

Speed	MIPI Rate B
Gear1	Up to 1.45Gbps
Note: Refer to the MIPI* M-PHY specifications for the additional information.	

Figure 10-8. USB Rise and Fall Times

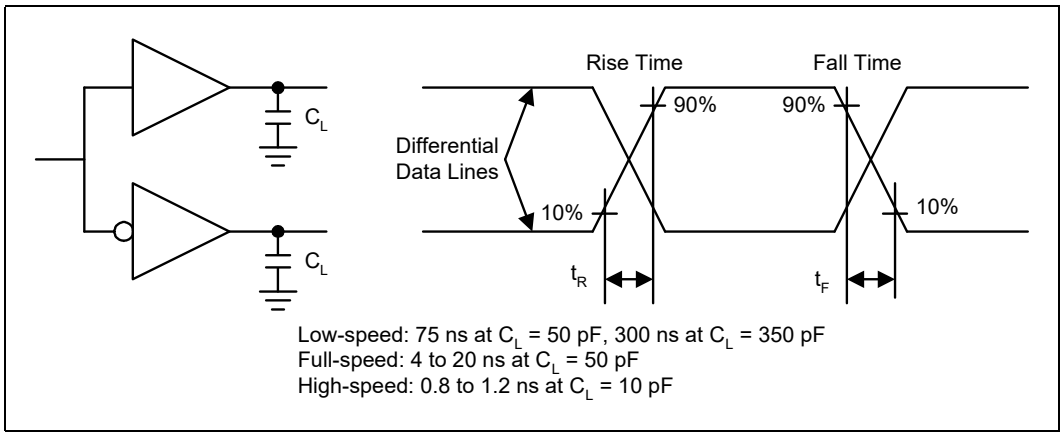


Figure 10-9. USB Jitter

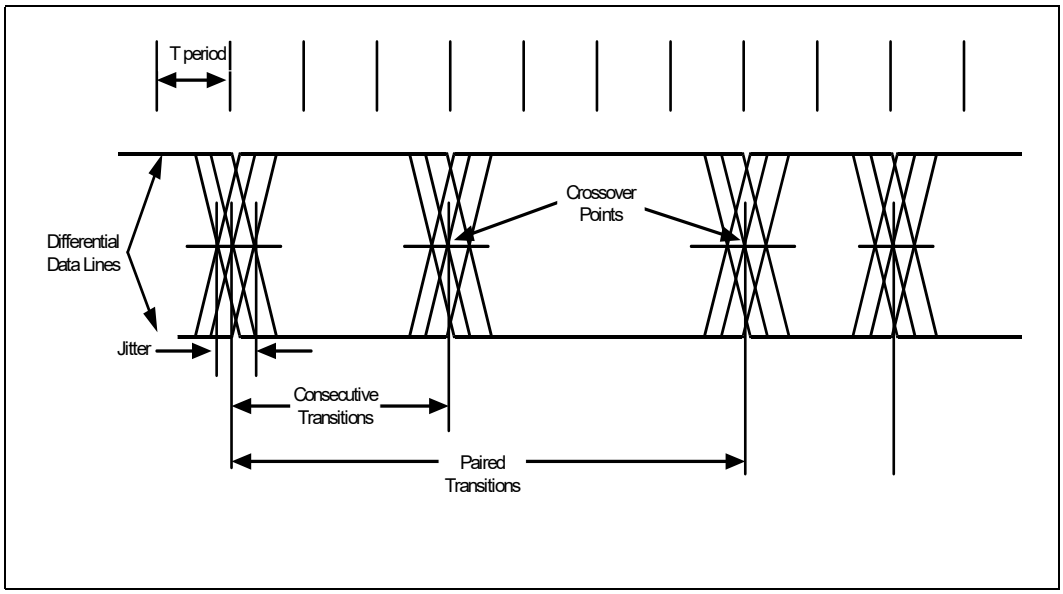


Figure 10-10.USB EOP Width

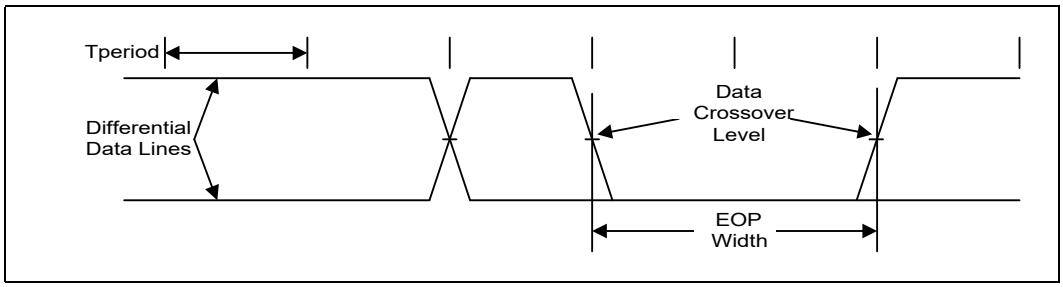


Table 10-17. SATA Interface Timings (Sheet 1 of 2)

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
UI-3	Gen III Operating Data Period (6Gb/s)	166.6083	166.6667	ps		

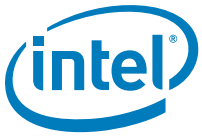


Table 10-17. SATA Interface Timings (Sheet 2 of 2)

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
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 100mV differential crosspoints of last and first edges of burst
4. Operating data period during Out-Of-Band burst transmissions

Table 10-18. SMBus and SMLink Timing (Sheet 1 of 2)

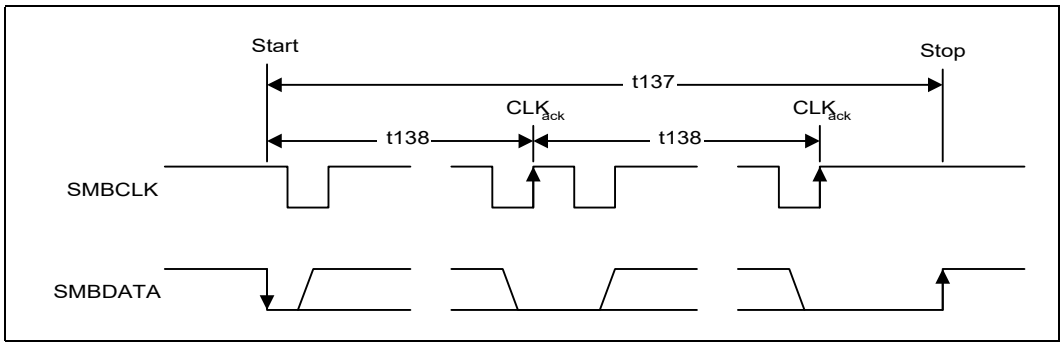
Sym	Parameter	Min.	Max.	Units	Notes	Figure
t130	Bus Free Time Between Stop and Start Condition	4.7	—	µs		10-6
t130SMLFM	Bus Free Time Between Stop and Start Condition	1.3	—	µs	5	10-6
t130SMLFMP	Bus Free Time Between Stop and Start Condition	0.5	—	µs	5	10-6
t131	Hold Time after (repeated) Start Condition. After this period, the first clock is generated.	4.0	—	µs		10-6
t131SMLFM	Hold Time after (repeated) Start Condition. After this period, the first clock is generated.	0.6	—	µs	5	10-6
t131SMLFMP	Hold Time after (repeated) Start Condition. After this period, the first clock is generated.	0.26	—	µs	5	10-6
t132	Repeated Start Condition Setup Time	4.7	—	µs		10-6
t132SMLFM	Repeated Start Condition Setup Time	0.6	—	µs	5	10-6
t132SMLFMP	Repeated Start Condition Setup Time	0.26	—	µs	5	10-6
t133	Stop Condition Setup Time	4.0	—	µs		10-6
t133SMLFM	Stop Condition Setup Time	0.6	—	µs	5	10-6
t133SMLFMP	Stop Condition Setup Time	0.26	—	µs	5	10-6
t134	Data Hold Time	300	—	ns	4	10-6
t134SMLFM	Data Hold Time	0	—	ns	4, 5	10-6
t134SMLFMP	Data Hold Time	0	—	ns	4, 5	10-6
t135	Data Setup Time	250	—	ns		10-6
t135SMLFM	Data Setup Time	100	—	ns	5	10-6
t135SMLFMP	Data Setup Time	50	—	ns	5	10-6
t136	Device Time Out	25	35	ms	1	
t137	Cumulative Clock Low Extend Time (secondary device)	—	25	ms	2	10-11
t138	Cumulative Clock Low Extend Time (primary device)	—	10	ms	3	10-11

Table 10-18. SMBus and SMLink Timing (Sheet 2 of 2)

Sym	Parameter	Min.	Max.	Units	Notes	Figure
T_{por}	Time in which a device must be operational after power-on reset	—	500	ms		

Notes:
1. A device will time out when any clock low exceeds this value.
2. t137 is the cumulative time a secondary device is allowed to extend the clock cycles in one message from the initial start to stop. If a secondary 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 primary 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.
4. t134 has a minimum timing for I^2C of 0 ns, while the minimum timing for SMBus/SMLINK is 300 ns.
5. Timings with the SMLFM designator apply only to SMLink[1,0] when operating in Fast Mode.

Figure 10-11. SMBus/SMLink Timeout



Note: SMBCLK also refers to SML[1:0]CLK and SMBDATA also refers to SML[1:0]DATA in Figure 10-6.

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

Symbol	Parameter	Min.	Max.	Units	Notes	Figure
t143	Time duration for which HDA_SDO is valid before HDA_BCLK edge.	7	—	ns		10-12
t144	Time duration for which HDA_SDO is valid after HDA_BCLK edge.	7	—	ns		10-12
t145	Setup time for HDA_SDI[1:0] at rising edge of HDA_BCLK	15	—	ns		10-12
t146	Hold time for HDA_SDI[1:0] at rising edge of HDA_BCLK	0	—	ns		10-12

Figure 10-12.Intel® High Definition Audio (Intel® HD Audio) Input and Output Timings

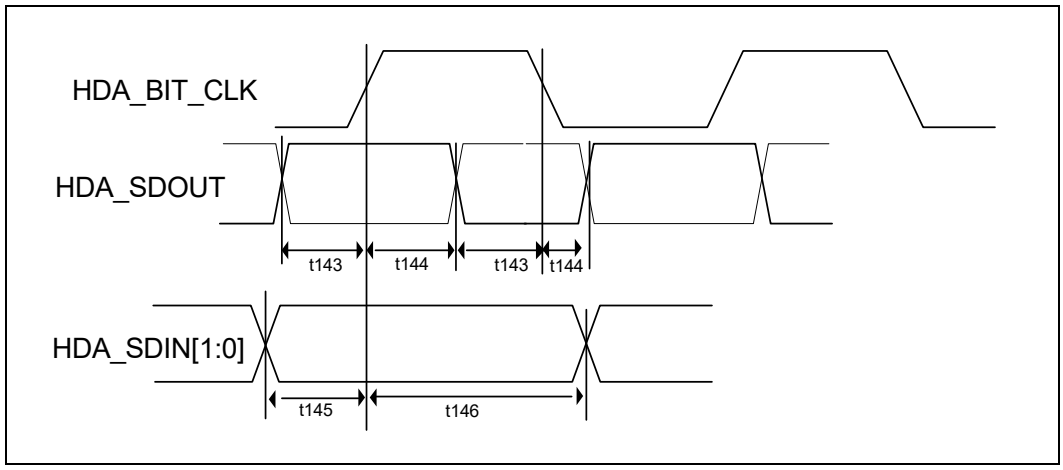


Table 10-20. LPC Timing

Sym	Parameter	Min.	Max.	Units	Notes	Figure
t150	LAD[3:0] Valid Delay from CLKOUT_LPC[1:0] Rising	3	24.67	ns		10-13
t151	LAD[3:0] Output Enable Delay from CLKOUT_LPC[1:0] Rising	2	—	ns		10-16
t152	LAD[3:0] Float Delay from CLKOUT_LPC[1:0] Rising	—	28	ns		10-15
t153	LAD[3:0] Setup Time to CLKOUT_LPC[1:0] Rising	17.67	—	ns		10-14
t154	LAD[3:0] Hold Time from CLKOUT_LPC[1:0] Rising	2	—	ns		10-14
t157	LFRAME# Valid Delay from CLKOUT_LPC[1:0] Rising	3	24.67	ns		10-13
Notes:	VT - Voltage threshold in input/output. It is a system generated spec that used to calculate all the related timing specs such as valid delay, pulse width, setup time, holding time, output enable delay, and float delay.					

Figure 10-13.Valid Delay from Rising Clock Edge

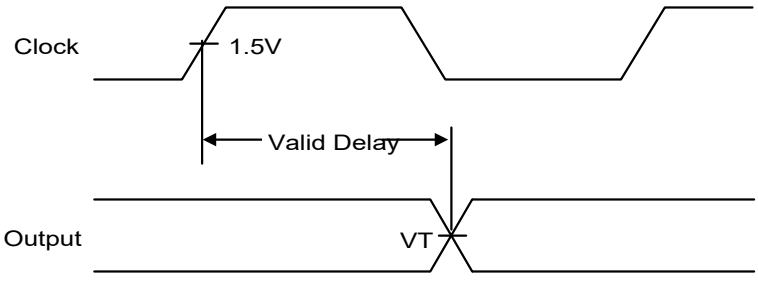


Figure 10-14.Set up and Hold Times

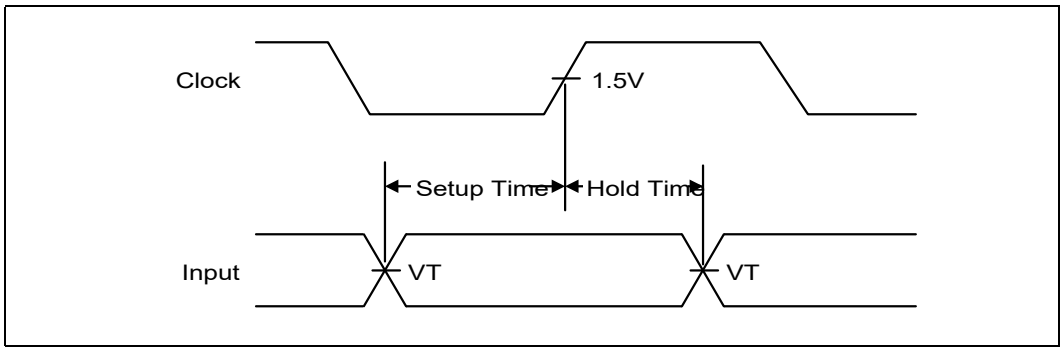


Figure 10-15.Float Delay

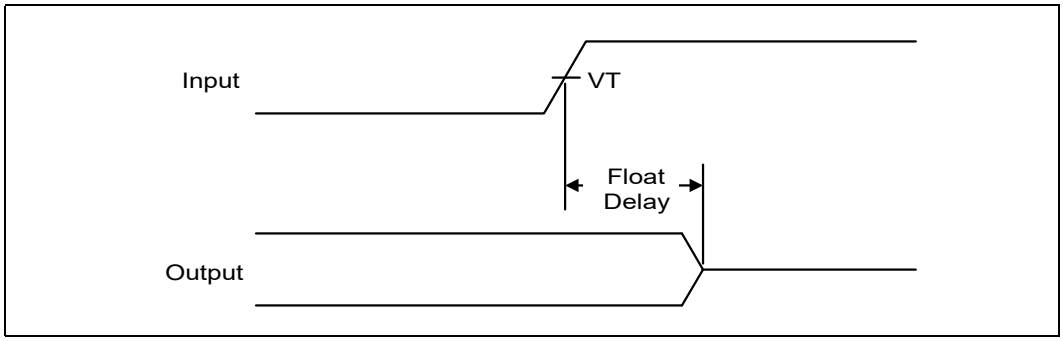


Figure 10-16.Output Enable Delay

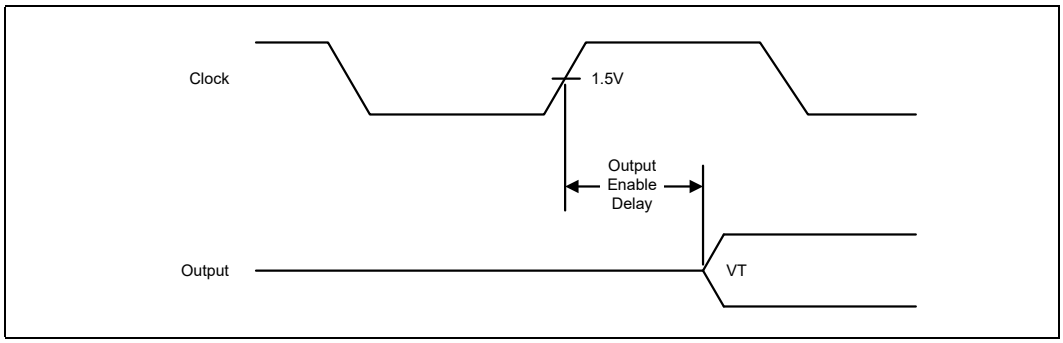


Table 10-21. Miscellaneous Timings

Symbol	Parameter	Min.	Max..	Units	Notes	Figure
t160	SERIRQ Setup Time to PCICLK Rising	7	—	ns		10-18
t161	SERIRQ Hold Time from PCICLK Rising	0	—	ns		
t162	GPIO, USB Resume Pulse Width	2	—	RTCCLK		10-19
t163	SPKR Valid Delay from OSC Rising	—	200	ns		10-17

Figure 10-17.Valid Delay from Rising Clock Edge

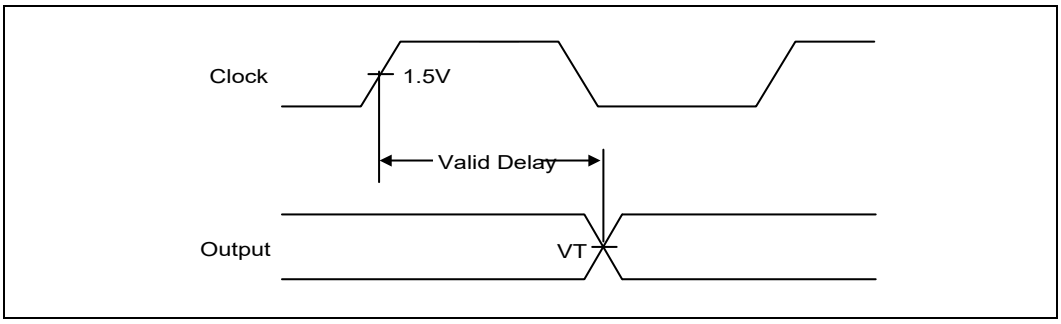


Figure 10-18.Set up and Hold Times

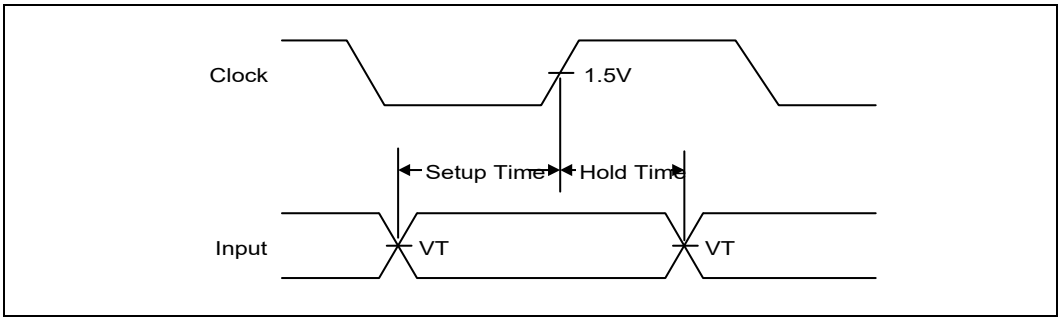


Figure 10-19.Pulse Width

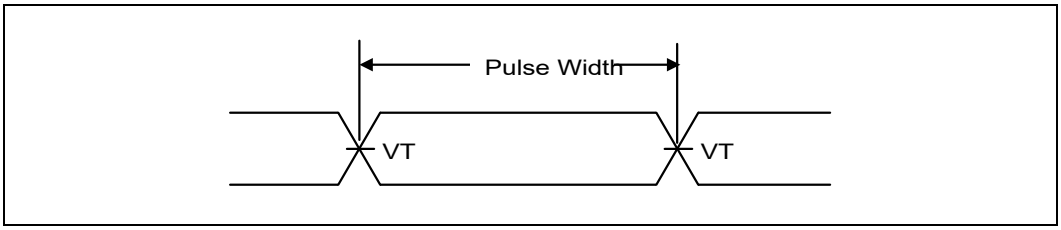


Table 10-22. SPI Timings (17MHz)

Symbol	Parameter	Min.	Max.	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	-5	13	ns		10-20
t184a	Setup of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	16	—	ns		10-20
t185a	Hold of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	0	—	ns		10-20
t186a	Setup of SPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		10-20
t187a	Hold of SPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		10-20
t188a	SPI CLK High time	26.37	—	ns		10-20
t189a	SPI CLK Low time	26.82	—	ns		10-20

Notes:
1. The typical clock frequency driven by the PCH is 17.86 MHz.
2. Measurement point for low time and high time is taken at 0.5(VccSPI).

Table 10-23. SPI Timings (30 MHz)

Symbol	Parameter	Min.	Max.	Units	Notes	Figure
t180b	Serial Clock Frequency	29.4	30.6	MHz	1	
t183b	Tco of SPI MOSI and SPI I/O with respect to serial clock falling edge at the host	-5	5	ns		10-20
t184b	Setup of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	8	—	ns		10-20
t185b	Hold of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	0	—	ns		10-20
t186b	Setup of SPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		10-20
t187b	Hold of SPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		10-20
t188b	SPI CLK High time	14.88	—	ns		10-20
t189b	SPI CLK Low time	15.18	—	ns		10-20

Note:
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).

Table 10-24. SPI Timings (48 MHz)

Symbol	Parameter	Min.	Max.	Units	Notes	Figure
t180c	Serial Clock Frequency	47.04	48.96	MHz	1	
t183c	Tco of SPI MOSI and SPI I/O with respect to serial clock falling edge at the host	-3	3	ns		10-20
t184c	Setup of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	8	—	ns		10-20
t185c	Hold of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	0	—	ns		10-20
t186c	Setup of SPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		10-20
t187c	Hold of SPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		10-20
t188c	SPI CLK High time	7.1	—	ns	2, 3	10-20
t189c	SPI CLK Low time	11.17	—	ns	2, 3	10-20

Note:
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).

Figure 10-20.SPI Timings

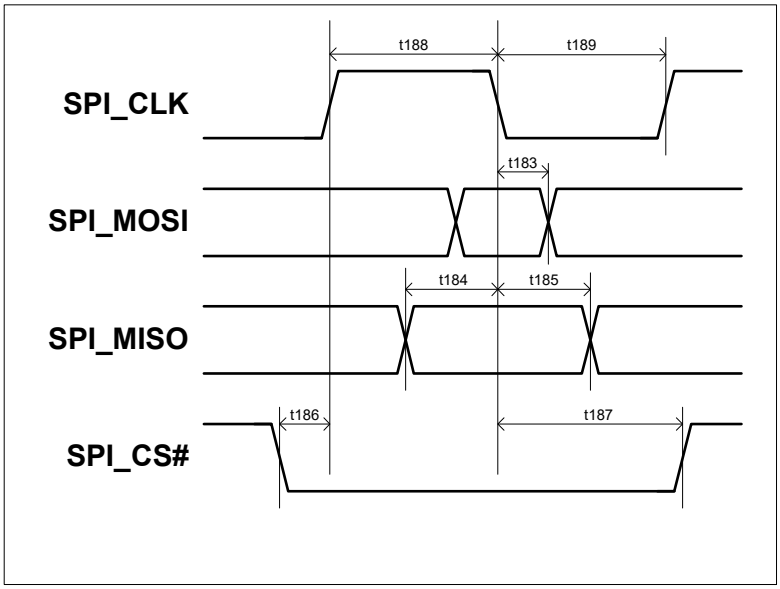


Table 10-25. GSPI Timings (20 MHz)

Symbol	Parameter	Min.	Max.	Units	Notes	Figure
F	Serial Clock Frequency		20	MHz		10-21
t183	Tco of SPI MOSI with respect to serial clock falling edge	-5	5	ns		10-21
t184	Setup of SPI MISO and SPI I/O with respect to serial clock rising edge	8		ns		10-21
t185	Hold of SPI MISO and SPI I/O with respect to serial clock rising edge	0		ns		10-21
t186	Setup of SPI CS# assertion with respect to serial clock rising edge	20		ns		10-21
t187	Hold of SPI CS# assertion with respect to serial clock falling edge	20		ns		10-21

Figure 10-21. GSPI Timings

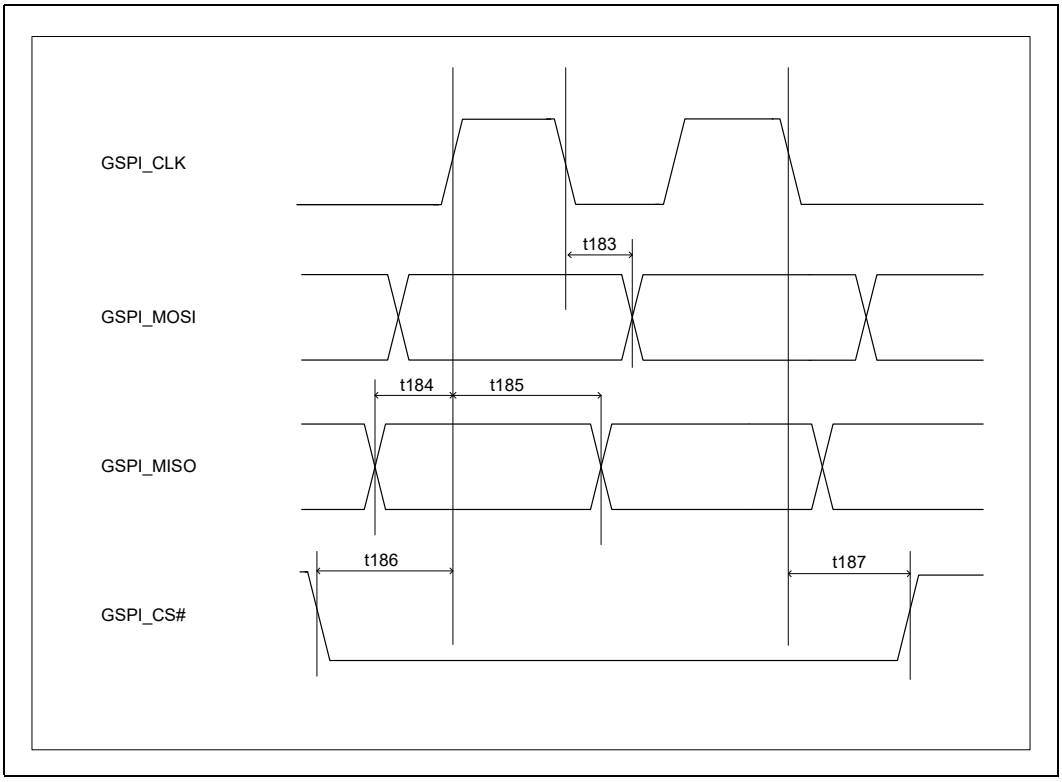


Table 10-26. Controller Link Receive Timings

Symbol	Parameter	Min.	Max..	Units	Notes	Figure
t190	Single-bit time	13	—	ns		10-22
t191	Single clock period	30	—	ns		10-22
t193	Setup time before CL_CLK	0.9	—	ns		10-22
t194	Hold time after CL_CLK	0.9	—	ns		10-22
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 10-22.Controller Link Receive Timings

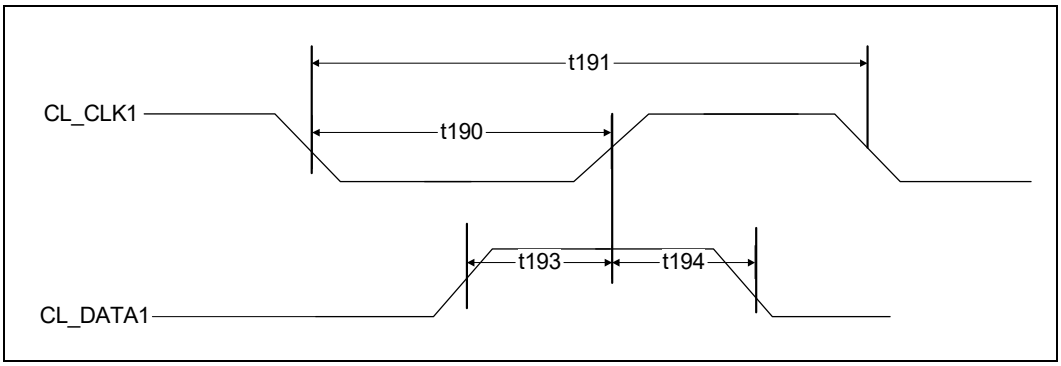


Figure 10-23.Controller Link Receive Slew Rate

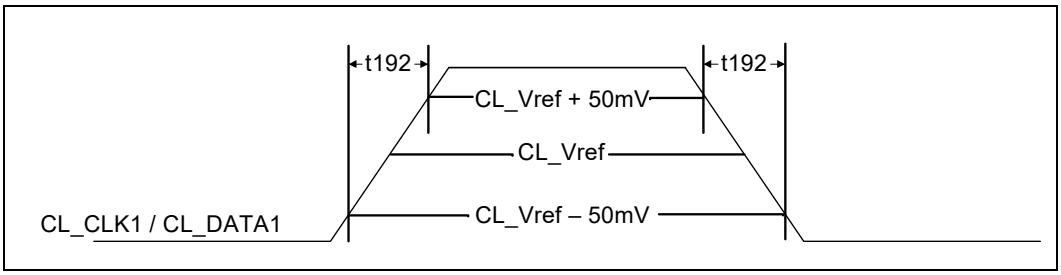


Table 10-27. UART Timings

Sym	Parameter	Min.	Max.	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 (Sheet 1 of 2)

Symbol	Parameter	Min.	Max.	Units	Notes	Figure
SCLK						
F_{I2S}	Clock Frequency in (Primary Mode)	-	8	MHz		
F_{I2S}	Clock Frequency (Secondary Mode)	-	9.6	MHz		
	Jitter	-	300	ps		
	Duty Cycle	45	55	%		
SFRM						
T_{CO}	Clock to Output Delay (PCH Primary Mode)	-	19	ns		
T_{INV}	SCLK edge to SFRM Invalid (PCH Primary Mode)	-	39	ns		
T_{SU}	Setup Time (PCH Secondary Mode)	19	-	ns		
T_{HD}	Hold Time (PCH Secondary Mode)	39	-	ns		
RXD						
T_{SU}	Setup Time (PCH Primary Mode)	14	-	ns		

Table 10-28. I^2S Timings (Sheet 2 of 2)

Symbol	Parameter	Min.	Max.	Units	Notes	Figure
T_{HD}	Hold Time (PCH Primary Mode)	40	-	ns		
T_{SU}	Setup Time (PCH Secondary Mode)	19	-	ns		
T_{HD}	Hold Time (PCH Secondary Mode)	39	-	ns		
TXD						
T_{CO}	Clock to Output Delay (PCH Primary Mode)	-	29	ns		
T_{INV}	SCLK edge to TXD Invalid (PCH Primary Mode)	-	29	ns		
T_{CO}	Clock to Output Delay (PCH Secondary Mode)	-	24	ns		
T_{INV}	SCLK edge to TXD Invalid (PCH Secondary Mode)	-	30	ns		

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 10-29 and Table 10-30 will ensure reliable I/O performance for the lifetime of the PCH.

Table 10-29. 3.3V Overshoot/Undershoot Specifications

Buffer Type	Associated Signal Group	Maximum Overshoot	Overshoot Duration	Maximum Undershoot	Undershoot Duration	Notes
CFIO HSHV	GPPD, GPPE	$1.39*V_{ccx}$	$0.25*T_{CH}$	$-0.39*V_{ccx}$	$0.25*T_{CH}$	1, 2
CFIO I2C	GPPB, GPPC, GPPD, GPPE, GPPF	$1.39*V_{ccx}$	$0.25*T_{CH}$	$-0.39*V_{ccx}$	$0.25*T_{CH}$	1, 2
CFIO LSHV	HDA, DSW, GPPA, GPPB, GPPC, GPPD, GPPE, GPPF, SPI	$1.39*V_{ccx}$	$0.25*T_{CH}$	$-0.39*V_{ccx}$	$0.25*T_{CH}$	1, 2
USB 2.0	USB 2.0	$1.32*V_{ccx}$	$0.25*T_{CH}$	$-0.32*V_{ccx}$	$0.25*T_{CH}$	1, 2

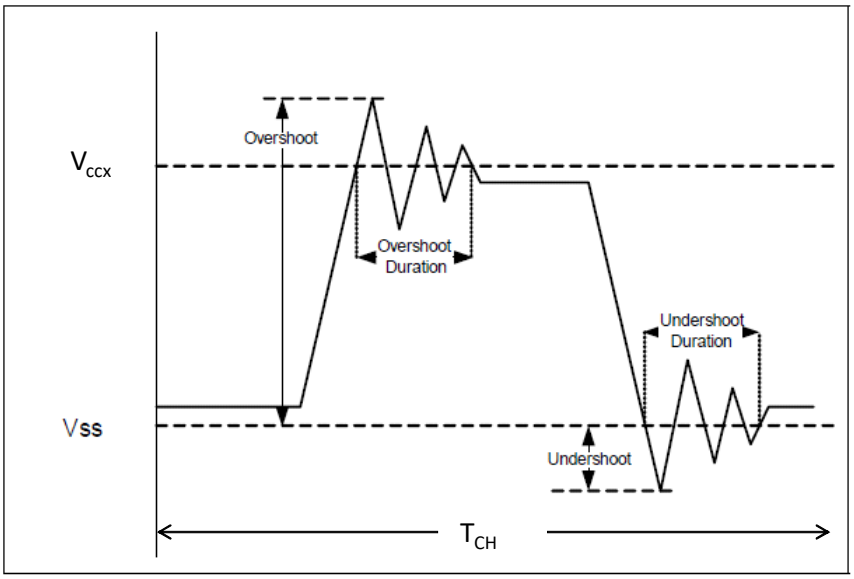
Notes:
1. These specifications are measured at the PCH pin.
2. Vccx refers to the supply voltage at the pin. TCH refers to the duration of the signal waveform. Refer to Figure 10-24 for pictorial description of allowable overshoot/undershoot magnitude and duration.

Table 10-30. 1.8V Overshoot/Undershoot Specifications

Buffer Type	Associated Signal Group	Maximum Overshoot	Overshoot Duration	Maximum Undershoot	Undershoot Duration	Notes
CFIO HSHV	GPPD, GPPE	2.57*V_{ccx}	0.25*T_{CH}	-1.57*V_{ccx}	0.25*T_{CH}	1, 2
CFIO I2C	GPPB, GPPC, GPPD, GPPE, GPPF	2.57*V_{ccx}	0.25*T_{CH}	-1.57*V_{ccx}	0.25*T_{CH}	1, 2
CFIO LSHV	HDA, DSW, GPPA, GPPB, GPPC, GPPD, GPPE, GPPF, SPI	2.57*V_{ccx}	0.25*T_{CH}	-1.57*V_{ccx}	0.25*T_{CH}	1, 2

Notes:
1. These specifications are measured at the PCH pin.
2. Vccx refers to the supply voltage at the pin. TCH refers to the duration of the signal waveform. Refer to Figure 10-24 for pictorial description of allowable overshoot/undershoot magnitude and duration.

Figure 10-24.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 the 8254 timers are in the core well. The 8254 unit is clocked by a 14.318-MHz clock derived from 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 (see NMI Status and Control ports).

11.1.1 Timer Programming

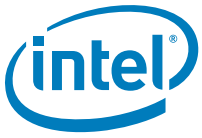
The counter/timers are programmed in the following fashion:

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.

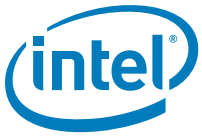
The Read Back command may be used to latch multiple counter outputs at one time. This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read or reprogrammed. Once read, a counter is unlatched. The 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 Integrated High Definition Audio

12.1 Acronyms

Acronyms	Description
DMIC	Digital Microphone Integrated Circuit
DSP	Digital Signal Processor
HDA	High Definition Audio
I^2S	Inter IC Sound
PCM	Pulse Code Modulation
SoC	System On Chip
VAD	Voice Activity Detector
VOIP	Voice Over Internet Protocol

12.2 References

None.

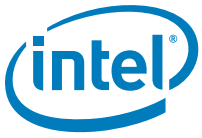
12.3 Overview

The Integrated High Definition Audio subsystem is a collection of controller, DSP, memory, and links that together can be used to provide a great platform audio experience. The controller, memory, and link form the basic audio controller to provide the streaming of audio from host software to an external audio codec with the host processor providing the audio enrichment. With the optional DSP enabled in the audio subsystem, it provides hardware acceleration for common audio and voice functions such as audio encode/decode, acoustic echo cancellation, noise cancellation, and so on. With such acceleration, the integration this integrated High Definition Audio subsystem in the PCH is expected to provide longer music playback times and VOIP call times for the platform.

12.4 Signal Description

Name	Type	Description
High Definition Audio Signals		
HDA_RST#/ I2S1_SCLK	O	**HD Audio Reset**: Primary H/W reset to internal/external codecs.
HDA_SYNC/ I2S0_SFRM	O	**HD Audio Sync:** 48-KHz fixed rate frame sync to the codecs. Also used to encode the stream number.
HDA_BCLK/ I2S0_SCLK	O	**HD Audio Bit Clock:** Up to 24-MHz serial data clock generated by the Intel HD Audio controller.
HDA_SDO/ I2S0_TXD	O	**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.

Name	Type	Description
HDA_SDI0/ I2S0_RXD	I	**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_RXD	I	**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/ HDA_BCLKGPP _D8	I/O	**I²S/PCM serial bit clock 0:** Clock used to control the timing of a transfer. Can be generated internally (Primary mode) or taken from an external source (Secondary mode).
I2S1_SCLK/ HDA_RST	I/O	**I²S/PCM serial bit clock 1:** This clock is used to control the timing of a transfer. Can be generated internally (Primary mode) or taken from an external source (Slave mode).
I2S2_SCLK/ GPP_F0	I/O	**I²S/PCM serial bit clock 2:** This clock is used to control the timing of a transfer. Can be generated internally (Primary mode) or taken from an external source (Secondary mode).
I2S0_SFRM/ GPP_D5/ HDA_SYNC	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 (Primary mode) or taken from an external source (Secondary mode).
I2S1_SFRM	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 (Primary mode) or taken from an external source (Secondary mode).
I2S2_SFRM/ GPP_F1	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 (Primary mode) or taken from an external source (Secondary mode).
I2S0_TXD // GPP_D6/ HDA_SDO	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	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/ GPP_F2	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/ GPP_D7/ HDA_SDI0	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/ HDA_SDI1	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/ GPP_F3	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/ GPP_D23	O	**I²S/PCM Primary reference clock:** This signal is the primary reference clock that connects to an audio codec.
DMIC Interface		
DMIC_CLK0/ GPP_D19	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_D17	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_D20	I	**Digital Mic Data:** Serial data input from the digital mic.
DMIC_DATA1 /GPP_D18	I	**Digital Mic Data:** Serial data input from the digital mic.



12.5 Integrated Pull-Ups and Pull-Downs

Table 12-1. Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value (Ω)	Notes
HDA_SYNC	Pull-down	9K-50K	
HDA_SDO	Pull-down	9K-50K	
HDA_SDI[1:0]	Pull-down	9K-50K	
I2S[2:0]_SFRM	Pull-Down	9K-50K	
I2S[2:0]_RXD	Pull-down	9K-50K	
DMIC_DATA[1:0]	Pull-down	9K-50K	

12.6 I/O Signal Planes and States

Table 12-2. I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately After Reset	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
I^2S/PCM Interface					
I2S0_SCLK	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S[2:1]_SCLK	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S0_SFRM	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S[2:1]_SFRM	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S0_TXD	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
I2S[2:1]_TXD	Primary	Driven Low	Driven Low	Driven Low	OFF
I2S0_RXD	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S[2:1]_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

12.7 Features

The Integrated High Definition Audio subsystem features are listed below.

12.7.1 High Definition Audio Controller Capabilities

- PCI/PCI Express controller
- Independent Primary Master logic for 16 general-purpose streams: 7 input and 9 output
- Supports variable length stream slots
- Supports up to:
  - 16 streams (7 input, 9 output)
  - 16 channels per stream
  - 32 bits/sample
  - 192 KHz sample rate
- Supports memory-based command/response transport
- 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)
  - Only if exposed as PCI Express device
- Supports Intel Power Optimizer Power Management
  - 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

12.7.2 Audio DSP Capabilities

- DSP offload for low-power audio rendering and recording
- Various DSP functions provided by Core: MP3, AAC, 3rd Party IP Algorithm, and so on
- Host downloadable DSP function module

12.7.3 High Definition Audio Link Capabilities

- 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

12.7.4 DSP I/O Peripherals Capabilities

- Two digital microphone ports to support up to four digital microphone modules
- Three bi-directional I^2S/PCM ports to support up to three I^2S connections

§ §

13 Controller Link

13.1 Overview

The Controller Link is used to manage the wireless LN device.

13.2 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 OD	**Controller Link Reset:** Controller Link reset that connects to a Wireless LAN Device supporting Intel Active Management Technology.

13.3 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value (Ohm)	Notes
CL_DATA	Pull-up Pull-down	31.25 100	See Section 13.4
CL_CLK	Pull-up Pull-down	31.25 100	See Section 13.4

13.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
CL_DATA	Primary	See Notes	See Notes	Internal Pull-down	Off
CL_CLK	Primary	See Notes	See 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.

13.5 Functional Description

The controller link is used to manage the wireless devices supporting Intel® Active Management Technology.

§ §

14 Processor Sideband Signals

14.1 Acronyms

Acronyms	Description
PECI	Platform Environmental Control Interface

14.2 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.
THERMTRIP#	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	Immediately after Reset	S3/S4/S5	Deep Sx
PROCPWRGD	Primary	Undriven[1]	Driven High	Off	Off
THERMTRIP#	Primary	Undriven	Undriven	Off	Off
PECI	Primary	Undriven	Undriven	Off	Off
CPU_GP[3:0]	Primary	Undriven	Undriven	Undriven	Off

Note: [1]Only when RSMRST is asserted low.

14.6 Functional Description

PROCPWRGD signal is outputs to the processor, to indicate 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 THERMTRIP# 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.

§ §

15 Digital Display Signals

15.1 Acronyms

Acronyms	Description
eDP*	embedded Display Port*

15.2 References

None

15.3 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 transcoder and analog display interface or port.

The PCH integrates digital display side band signals AUX CH, DDC bus, and Hot-Plug Detect signals even though digital display interfaces are moved to processor. There are two pairs of AUX CH, DDC Clock/Data, and Hot-Plug Detect signals on the PCH that correspond to digital display interface/ports.

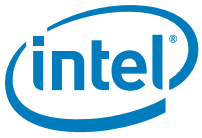
Auxiliary Channel (AUX CH) is a half-duplex bidirectional channel used for link management and device control. AUX CH is an AC coupled differential signal.

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 follows 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.

Table 15-1. Digital Display Signals

Name	Type	Description
DDPB_HPD0 / GPP_E13	I	**Display Port B:** HPD Hot-Plug Detect
DDPC_HPD1 / GPP_E14	I	**Display Port C:** HPD Hot-Plug Detect
DDPD_HPD2 / GPP_E15	I	**Display Port D:** HPD Hot-Plug Detect or eDP*[1] Hot-Plug Detect
DDPE_HPD3 / GPP_E16	I	**Display Port E:** HPD Hot-Plug Detect
DDPB_CTRLCLK / GPP_E18	I/O	**Display Port B:** Control Clock.
DDPB_CTRLDATA / GPP_E19	I/O	**Display Port B:** Control Data.
DDPC_CTRLCLK / GPP_E20	I/O	**Display Port C:** Control Clock
DDPC_CTRLDATA / GPP_E21	I/O	**Display Port C:** Control Data



15.4 Embedded DisplayPort* (eDP*) Backlight Control Signals

Name	Type	Description
eDP_VDDEN	O	**eDP* Panel Power Enable:** Panel power controls 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
EDP_HPD / GPP_E17	I	**eDP*:** Hot-Plug Detect
Note: eDP_VDDEN, eDP_BKLTEN, eDP_BKLTCTL can be left as no connect if eDP* is not used.		

15.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
DDPB_CTRLDATA	Pull-down	15K-40K	See note below
DDPC_CTRLDATA	Pull-down	15K-40K	See note below

Note: The internal pullup/pulldown 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.

15.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
DDPB_HPD0	Primary	Undriven	Undriven	Undriven	Off
DDPC_HPD1	Primary	Undriven	Undriven	Undriven	Off
DDPD_HPD2	Primary	Undriven	Undriven	Undriven	Off
DDPE_HPD3	Primary	Undriven	Undriven	Undriven	Off
DDPB_CTRLCLK	Primary	Undriven	Undriven	Undriven	Off
DDPB_CTRLDATA	Primary	Internal Pull-down	Driven Low	Internal Pull-down	Off
DDPC_CTRLCLK	Primary	Undriven	Undriven	Undriven	Off
DDPC_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
EDP_HPD	Primary	Undriven	Undriven	Undriven	Off

§ §

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 References

None.

16.3 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.

The interface supports 1.8V only and is a dedicated, single-secondary eSPI bus interface for client platforms. This interface is not shared and distinct from the SPI bus interface used for flash device and TPM.

Note: The PCH LPC and eSPI coexist but are mutually exclusive. A HW strap is used to determine which interface is used on the platform.

16.4 Signal Description

Name	Type	Description
ESPI_CLK/ CLKOUT_LPC0/ GPP_A9	O	**eSPI Clock:** eSPI clock output from the PCH to secondary device.
ESPI_IO0/ LAD0/GPP_A1	I/O	**eSPI Data Signal 0:** Bi-directional pin used to transfer data between the PCH and eSPI secondary device.
ESPI_IO1/ LAD1/GPP_A2	I/O	**eSPI Data Signal 1:** Bi-directional pin used to transfer data between the PCH and eSPI secondary device
ESPI_IO2/ LAD2/GPP_A3	I/O	**eSPI Data Signal 2:** Bi-directional pin used to transfer data between the PCH and eSPI secondary device
ESPI_IO3/ LAD3/GPP_A4	I/O	**eSPI Data Signal 3:** Bi-directional pin used to transfer data between the PCH and eSPI secondary device
ESPI_CS#/ LFRAME#/ GPP_A5	O	**eSPI Chip Select:** Driving CS# signal low to select eSPI secondary for the transaction.

Name	Type	Description
ESPI_RST#/ SUS_STAT#/ GPP_A14	O	**eSPI Reset:** Reset signal from the PCH to eSPI secondary.

16.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
ESPI_CLK	Pull-down	9K - 50K	
ESPI_IO[3:0]	Pull-up	15K - 40K	
ESPI_ CS #	Pull-up	15K - 40K	

16.6 I/O Signal Planes and States

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

16.7 Functional Description

16.7.1 Features

The PCH eSPI controller supports the following features:

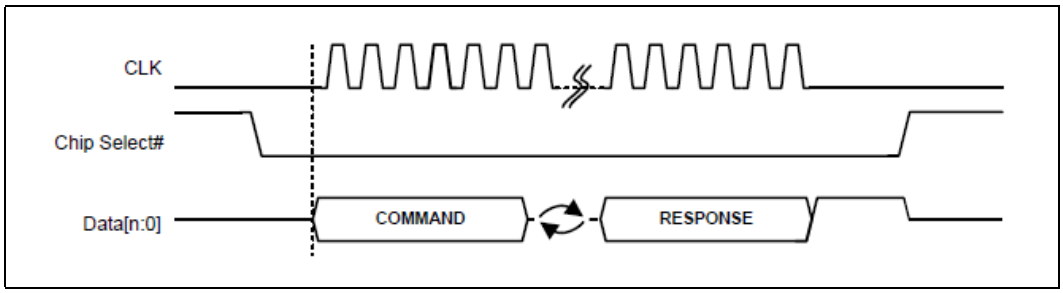
- Primary mode only, allowing one secondary device to be connected to the PCH
- Support for 20 MHz, 24 MHz, 30 MHz, 48 MHz, and 60 MHz (configured by soft straps)
- 1.8V support only
- Up to quad mode support
- In-band messages for communication between the PCH and secondary device to eliminate side-band signals
- Real-time SPI flash sharing, allowing real-time operational access by the PCH and secondary device
- Transmitting RTC time/date to the secondary device upon request

Note: For client platform, the PCH eSPI controller does not support a discrete ALERT# pin (as described in the eSPI specification) since the PCH supports only a Single Primary - Single Secondary configuration. Only ALERT# signaling (over ESPI_IO1) is supported.

16.7.2 Protocols

The following figure is an overview of the basic eSPI protocol.

Figure 16-1. Basic eSPI Protocol



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

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 secondary.

The serial clock must be low at the assertion edge of the CS# while ESPI_RST# 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 secondary. 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 secondary. Data from the secondary 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).

16.7.3 WAIT States from Secondary eSPI

There are situations when the secondary cannot predict the length of the command packet from the primary (PCH). For non-posted transactions, the secondary 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 secondary after the TAR cycles.

16.7.4 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 secondary 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 secondary.

16.7.5 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 secondary is connected to the interface. The controller does not perform any other discovery to confirm the presence of the secondary connection.

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

16.7.6 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 secondary eSPI is discovered through the GET_CONFIGURATION command issued by the PCH to the secondary eSPI during initialization.

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

Table 16-1. eSPI Channels and Supported Transactions

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
3	Flash Access	N/A	Flash Read, Write, Erase
N/A	General	Register Accesses	N/A

16.7.6.1 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 secondary eSPI (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 secondary eSPI to the Host. These include Memory Reads and Writes.

16.7.6.2 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 secondary eSPI to the PCH. Both edge and triggered interrupts are supported.

16.7.6.2.1 eSPI Virtual Wires (VW)

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

Table 16-2. eSPI Virtual Wires (VW)

Virtual Wire	PCH Pin Direction	Reset Control	Pin Retained in PCH (For Use by Other Components)
SUS_STAT#	Output	ESPI_RESET#	No
SUS_PWRDN_ACK	Output	ESPI_RESET#	No
PLTRST#	Output	ESPI_RESET#	Yes
PME#	Input	ESPI_RESET#	No
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
SLP_S3#/SLP_S4#/ SLP_S5#/SLP_LAN#/ SLP_WLAN#	Output	DSW_PWROK	Yes

16.7.6.2.2 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 secondary eSPI 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 secondary. It does not maintain interrupt state, polarity or enable for any of the interrupt events.

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

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

- Tunnel MCTP Packets between the Intel® ME and secondary eSPI device: The Intel ME 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 secondary 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 ME and eSPI device.
- Tunnel PCH Temperature Data to the secondary eSPI: The eSPI controller stores the PCH temperature data internally and sends it to the secondary using a posted OOB message when a request is made to a specific destination address.
- Tunnel PCH RTC Time and Date Bytes to the secondary eSPI: the eSPI controller captures this data internally at periodic intervals from the PCH RTC controller and sends it to the secondary device using a posted OOB message when a request is made to a specific destination address.

16.7.6.3.1 PCH Temperature Data Over eSPI OOB Channel

eSPI controller supports the transmitting of PCH thermal data to the secondary eSPI. 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 secondary eSPI and the PCH response back are shown in Figure 16-2 and Figure 16-3.

Figure 16-2. Secondary eSPI Request to PCH 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 Secondary eSPI 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]							

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

The PCH eSPI controller supports the transmitting of PCH RTC time/date to the secondary eSPI. This allows the secondary eSPI 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 secondary eSPI and the PCH response back to the device are shown in Figure 16-4 and Figure 16-5.

Figure 16-4. Secondary eSPI Request to PCH 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 Secondary eSPI with RTC Time

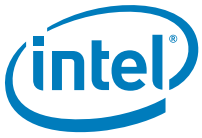
Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message = 21h							
1	Tag[3:0]				Length[11:8] = 0h			
2	Length[7:0] = 0Ch							
3	Destination Slave Addr. = 0Fh (eSPI Slave 0 [EC])							0
4	Command Code = 02h (Get_PCH_RTC_Time)							
5	Byte Count = 09h							
6	Source Slave Addr. = 02h (PCH OOB HW 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.

16.7.6.4 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 secondary eSPI 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 secondary eSPI in a separate completion packet.

16.7.6.4.1 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.

§ §

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 References

None

17.3 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. Each of these pin groups has a dedicated power pin that can be set to either 1.8V or 3.3V. All pins within the same group (including the native functionality that is multiplexed with the GPIO) operate at the same voltage determined by the power supplied to the power pins.

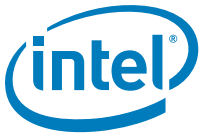
All PCH GPIOs can be configured as input or output signals. Many GPIOs are multiplexed with other functions.

SCI and IOxAPIC interrupt capability is available on all GPIOs. NMI and SMI capability is available on selected GPIOs only.

Many GPIOs are glitch-free.

Table 17-1. GPIO Group Summary

GPIO Group	Power Pins	Voltage
Primary Well Group A (GPP_A)	VCCPGPPA	1.8V or 3.3V
Primary Well Group B (GPP_B)	VCCPGPPB	1.8V or 3.3V
Primary Well Group C (GPP_C)	VCCPGPPC	1.8V or 3.3V
Primary Well Group D (GPP_D)	VCCPGPPD	1.8V or 3.3V
Primary Well Group E (GPP_E)	VCCPGPPE	1.8V or 3.3V
Primary Well Group F (GPP_F)	VCCPGPPF	1.8V
Primary Well Group G (GPP_G)	VCCPGPPG	1.8V or 3.3V
Deep Sleep Well Group (GPD)	VCCPDSW_3p3	3.3V



17.4 Signal Description

Table 17-2 summarizes the GPIO implementation in the PCH.

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

Name	Internal Pull-Up/ Pull-Down (Note 1)	De-Glitch (Note 2)		Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 3)	Default	NMI or SMI Capable	Notes
		Input	Output				
Group A GPIO - Primary Power Well (1.8V or 3.3V)							
GPP_A0	None	No	Yes (Note 6)	LPC Mode: RCIN#(1st) eSPI Mode: none	RCIN# (LPC mode) GPI (eSPI mode)	None	See Note 5
GPP_A1	None	No	Yes (Note 6)	LPC mode: LAD0 (1st) eSPI mode: ESPI_IO0 (3rd)	LAD0 (LPC mode) ESPI_IO0 (eSPI mode)	None	See Note 5
GPP_A2	None	No	Yes (Note 6)	LPC mode: LAD1 (1st) eSPI mode: ESPI_IO1 (3rd)	LAD1 (LPC mode) ESPI_IO1 (eSPI mode)	None	See Note 5
GPP_A3	None	No	Yes (Note 6)	LPC mode: LAD2 (1st) eSPI mode: ESPI_IO2 (3rd)	LAD2 (LPC mode) ESPI_IO2 (eSPI mode)	None	See Note 5
GPP_A4	None	No	Yes (Note 6)	LPC mode: LAD3 (1st) eSPI mode: ESPI_IO3 (3rd)	LAD3 (LPC mode) ESPI_IO3 (eSPI mode)	None	See Note 5
GPP_A5	None	No	Yes (Note 7)	LPC mode: LFRAME# (1st) eSPI mode: ESPI_CS# (3rd)	LFRAME# (LPC mode) ESPI_CS# (eSPI mode)	None	See Note 5
GPP_A6	None	No	Yes (Note 6)	LPC Mode: SERIRQ (1st) eSPI Mode: none	SERIRQ (LPC mode) GPI (eSPI mode)	None	See Note 5
GPP_A7	None	No	Yes (Note 6)	LPC Mode: PIRQA# (1st) eSPI Mode: none	PIRQA# (LPC mode) GPI (eSPI mode)	None	See Note 5
GPP_A8	None	No	Yes (Note 6)	LPC Mode: CLKRUN# (1st) eSPI Mode: None	CLKRUN# (LPC mode) GPI (eSPI mode)	None	See Note 5
GPP_A9	None	No	Yes (Note 6)	LPC mode: CLKOUT_LPC0 (1st) eSPI mode: ESPI_CLK (3rd)	CLKOUT_LPC0 (LPC Mode) ESPI_CLK (eSPI mode)	None	See Note 5
GPP_A10	None	No	Yes (Note 6)	LPC Mode: CLKOUT_LPC1 eSPI Mode: None	CLKOUT_LPC1 (LPC mode) GPI (eSPI mode)	None	See Note 5
GPP_A11	None	No	Yes (Note 6)	LPC mode: PME# eSPI mode: None	PME# (LPC mode) GPI (eSPI mode)	None	See Note 5

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

Name	Internal Pull-Up/ Pull-Down (Note 1)	De-Glitch (Note 2)		Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 3)	Default	NMI or SMI Capable	Notes
		Input	Output				
GPP_A12	None	No	Yes (Note 6)	BM_BUSY# (1st)/ ISH_GP6 (2nd) / SX_EXIT_HOLDOFF# (3rd)	GPI	None	
GPP_A13	None	No	Yes (Note 7)	LPC mode: SUSWARN# / SUSPWRDNACK eSPI mode: None	SUSWANRN#/ SUSPWRDNACK (LPC mode) GPI (eSPI mode)	None	See Note 5
GPP_A14	None	No	Yes (Note 7)	LPC mode: SUS_STAT# (1st) eSPI mode: ESPI_RESET# (3rd)	SUS_STAT# (LPC mode) ESPI_RESET# (eSPI mode)	None	See Note 5
GPP_A15	None	No	Yes (Note 6)	LPC mode: SUS_ACK# eSPI mode: None	SUS_ACK# (LPC mode) GPI (eSPI mode)	None	See Note 5
GPP_A16	None	No	Yes (Note 6)	SD_1P8_SEL	GPI	None	
GPP_A17	None	No	Yes (Note 6)	SD_PWR_EN# (1st)/ ISH_GP7 (2nd)	GPI	None	
GPP_A18	None	No	Yes (Note 6)	ISH_GP0	GPI	None	
GPP_A19	None	No	Yes (Note 6)	ISH_GP1	GPI	None	
GPP_A20	None	No	Yes (Note 6)	ISH_GP2	GPI	None	
GPP_A21	None	No	Yes (Note 6)	ISH_GP3	GPI	None	
GPP_A22	None	No	Yes (Note 6)	ISH_GP4	GPI	None	
GPP_A23	None	No	Yes (Note 6)	ISH_GP5	GPI	None	
Group B GPIO - Primary Power Well (1.8V or 3.3V)							
GPP_B0	None	No	Yes (Note 7)	CORE_VID0	CORE_VID0	None	
GPP_B1	None	No	Yes (Note 7)	CORE_VID1	CORE_VID1	None	
GPP_B2	None	No	Yes (Note 6)	VRALERT#	GPI	None	
GPP_B3	None	No	Yes (Note 6)	CPU_GP2	GPI	None	
GPP_B4	None	No	Yes (Note 6)	CPU_GP3	GPI	None	
GPP_B5	None	No	Yes (Note 6)	SRCCLKREQ0#	GPI	None	
GPP_B6	None	No	Yes (Note 6)	SRCCLKREQ1#	GPI	None	
GPP_B7	None	No	Yes (Note 6)	SRCCLKREQ2#	GPI	None	

Table 17-2. General Purpose I/O Signals (Sheet 3 of 9)

Name	Internal Pull-Up/ Pull-Down (Note 1)	De-Glitch (Note 2)		Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 3)	Default	NMI or SMI Capable	Notes
		Input	Output				
GPP_B8	None	No	Yes (Note 6)	SRCCLKREQ3#	GPI	None	
GPP_B9	None	No	Yes (Note 6)	SRCCLKREQ4#	GPI	None	
GPP_B10	None	No	Yes (Note 6)	SRCCLKREQ5#	GPI	None	
GPP_B11	None	No	Yes (with weak PU)	EXT_PWR_GATE#	EXT_PWR_GATE#	None	
GPP_B12	None	No	Yes (Note 9)	SLP_S0#	SLP_S0#	None	
GPP_B13	None	No	Yes (Note 7)	PLTRST#	PLTRST#	None	
GPP_B14	20K PD (Note 4)	No	No	SPKR	GPO	NMI SMI	•Also used as a strap. •The Pull-down resistor is disabled after RSMRST# de-asserts •As GPO, the signal defaults to '0
GPP_B15	None	No	Yes (Note 6)	GSPI0_CS#	GPI	None	
GPP_B16	None	No	Yes (Note 6)	GSPI0_CLK	GPI	None	
GPP_B17	None	No	Yes (Note 6)	GSPI0_MISO	GPI	None	
GPP_B18	20K PD (Note 4)	No	No	GSPI0_MOSI	GPO	None	•Also used as a strap. •The Pull-down resistor is disabled after PCH_PWROK de-asserts •As GPO, the signal defaults to '0
GPP_B19	None	No	Yes (Note 6)	GSPI1_CS#	GPI	None	
GPP_B20	None	No	Yes (Note 6)	GSPI1_CLK	GPI	NMI SMI	
GPP_B21	None	No	Yes (Note 6)	GSPI1_MISO	GPI	None	
GPP_B22	20K PD (Note 4)	No	No	GSPI1_MOSI	GPO	None	•Also used as a strap. •The Pull-down resistor is disabled after PCH_PWROK de-asserts •As GPO, the signal defaults to '0
GPP_B23	20K PD (Note 4)	Yes (Note 8)	No	SML1ALERT# (1st)/ PCHHOT# (2nd)	GPO	NMI SMI	•Also used as a strap. •The Pull-down resistor is disabled after RSMRST# de-asserts •As GPO, the signal defaults to '0

Table 17-2. General Purpose I/O Signals (Sheet 4 of 9)

Name	Internal Pull-Up/ Pull-Down (Note 1)	De-Glitch (Note 2)		Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 3)	Default	NMI or SMI Capable	Notes
		Input	Output				
Group C GPIO - Primary Power Well (1.8V or 3.3V)							
GPP_C0	None	No	Yes (Note 6)	SMBCLK	SMBCLK	None	
GPP_C1	None	No	Yes (Note 6)	SMBDATA	SMBDATA	None	
GPP_C2	20K PD (Note 4)	No	No	SMBALERT#	GPO	None	•Also used as a strap. •The Pull-down resistor is disabled after RSMRST# de-asserts •As GPO, the signal defaults to `0
GPP_C3	None	No	Yes (Note 6)	SML0CLK	SML0CLK	None	
GPP_C4	None	No	Yes (Note 6)	SML0DATA	SML0DATA	None	
GPP_C5	20K PD (Note 4)	No	No	SML0ALERT#	GPO	None	•Also used as a strap. •The Pull-down resistor is disabled after RSMRST# de-asserts •As GPO, the signal defaults to `0
GPP_C6	None	No	Yes (Note 6)	SML1CLK	GPI	None	
GPP_C7	None	No	Yes (Note 6)	SML1DATA	GPI	None	
GPP_C8	None	No	Yes (Note 6)	UART0_RXD	GPI	None	
GPP_C9	None	No	Yes (Note 6)	UART0_TXD	GPI	None	
GPP_C10	None	No	Yes (Note 6)	UART0_RTS#	GPI	None	
GPP_C11	None	No	Yes (Note 6)	UART0_CTS#	GPI	None	
GPP_C12	None	No	Yes (Note 6)	UART1_RXD (1st)/ ISH_UART1_RXD (2nd)	GPI	None	
GPP_C13	None	No	Yes (Note 6)	UART1_TXD (1st) / ISH_UART1_TXD (2nd)	GPI	None	
GPP_C14	None	No	Yes (Note 6)	UART1_RTS# (1st) / ISH_UART1_RTS# (2nd)	GPI	None	
GPP_C15	None	No	Yes (Note 6)	UART1_CTS# (1st) / ISH_UART1_CTS# (2nd)	GPI	None	
GPP_C16	None	Yes (Note 8)	Yes (Note 6)	I2C0_SDA	GPI	None	
GPP_C17	None	Yes (Note 8)	Yes (Note 6)	I2C0_SCL	GPI	None	

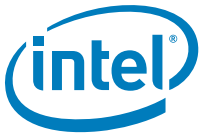


Table 17-2. General Purpose I/O Signals (Sheet 5 of 9)

Name	Internal Pull-Up/ Pull-Down (Note 1)	De-Glitch (Note 2)		Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 3)	Default	NMI or SMI Capable	Notes
		Input	Output				
GPP_C18	None	Yes (Note 8)	Yes (Note 6)	I2C1_SDA	GPI	None	
GPP_C19	None	Yes (Note 8)	Yes (Note 6)	I2C1_SCL	GPI	None	
GPP_C20	None	No	Yes (Note 6)	UART2_RXD	GPI	None	
GPP_C21	None	No	Yes (Note 6)	UART2_TXD	GPI	None	
GPP_C22	None	No	Yes (Note 6)	UART2_RTS#	GPI	NMI SMI	
GPP_C23	None	No	Yes (Note 6)	UART2_CTS#	GPI	NMI SMI	
Group D GPIO in Primary Power Well (1.8V or 3.3V)							
GPP_D0	None	No	Yes (Note 6)	None	GPI	NMI SMI	This GPIO is blink capable
GPP_D1	None	No	Yes (Note 6)	None	GPI	NMI SMI	This GPIO is blink capable
GPP_D2	None	No	Yes (Note 6)	None	GPI	NMI SMI	This GPIO is blink capable
GPP_D3	None	No	Yes (Note 6)	None	GPI	NMI SMI	This GPIO is blink capable
GPP_D4	None	No	Yes (Note 6)	FLASHTRIG	GPI	NMI SMI	This GPIO is blink capable
GPP_D5	None	Yes (Note 8)	Yes (Note 6)	ISH_I2C0_SDA	GPI	None	
GPP_D6	None	Yes (Note 8)	Yes (Note 6)	ISH_I2C0_SCL	GPI	None	
GPP_D7	None	Yes (Note 8)	Yes (Note 6)	ISH_I2C1_SDA	GPI	None	
GPP_D8	None	Yes (Note 8)	Yes (Note 6)	ISH_I2C1_SCL	GPI	None	
GPP_D9	None	No	Yes (Note 6)	None	GPI	None	
GPP_D10	None	No	Yes (Note 6)	None	GPI	None	
GPP_D11	None	No	Yes (Note 6)	None	GPI	None	
GPP_D12	None	No	Yes (Note 6)	None	GPI	None	
GPP_D13	None	No	Yes (Note 6)	ISH_UART0_RXD	GPI	None	
GPP_D14	None	No	Yes (Note 6)	ISH_UART0_TXD	GPI	None	
GPP_D15	None	No	Yes (Note 6)	ISH_UART0_RTS#	GPI	None	
GPP_D16	None	No	Yes (Note 6)	ISH_UART0_CTS#	GPI	None	

Table 17-2. General Purpose I/O Signals (Sheet 6 of 9)

Name	Internal Pull-Up/ Pull-Down (Note 1)	De-Glitch (Note 2)		Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 3)	Default	NMI or SMI Capable	Notes
		Input	Output				
GPP_D17	None	No	Yes (Note 6)	DMIC_CLK1	GPI	None	
GPP_D18	None	No	Yes (Note 6)	DMIC_DATA1	GPI	None	
GPP_D19	None	No	Yes (Note 6)	DMIC_CLK0	GPI	None	
GPP_D20	None	No	Yes (Note 6)	DMIC_DATA0	GPI	None	
GPP_D21	None	No	Yes (Note 6)	None	GPI	None	
GPP_D22	None	No	Yes (Note 6)	None	GPI	None	
GPP_D23	None	No	Yes (Note 6)	I2S_MCLK	GPI	None	
Group E GPIO - Primary Power Well (1.8V or 3.3V)							
GPP_E0	None	No	Yes (Note 6)	SATAXPCIE0 (1st)/ SATAGP0 (2nd)	SATAXPCIE0 / SATAGP0 or GPI	NMI SMI	Default SATAXPCIE0 is set by a soft strap. Default is GPI before soft straps are loaded
GPP_E1	None	No	Yes (Note 6)	SATAXPCIE1 1st / SATAGP1 (2nd)	SATAXPCIE1/ SATAGP1 or GPI	NMI SMI	Default SATAXPCIE1 is set by a soft strap. Default is GPI before soft straps are loaded
GPP_E2	None	No	Yes (Note 6)	SATAXPCIE2 (1st) /SATAGP2 (2nd)	SATAXPCIE2/ SATAGP2 or GPI	NMI SMI	Default SATAXPCIE2 is set by a soft strap. Default state is GPI before soft straps are loaded
GPP_E3	None	No	Yes (Note 6)	CPU_GP0	GPI	NMI SMI	
GPP_E4	None	No	Yes (Note 6)	SATA_DEVSLP0	GPI	NMI SMI	
GPP_E5	None	No	Yes (Note 6)	SATA_DEVSLP1	GPI	NMI SMI	
GPP_E6	None	No	Yes (Note 6)	SATA_DEVSLP2	GPI	NMI SMI	
GPP_E7	None	No	Yes (Note 6)	CPU_GP1	GPI	NMI SMI	
GPP_E8	None	No	Yes (Note 6)	SATALED#	GPI	NMI SMI	
GPP_E9	20K PD (Note 4)	No	Yes (Note 6)	USB2_OC0#	GPI	None	•The Pull-down resistor is disabled after RSMRST# de-asserts
GPP_E10	20K PD (Note 4)	No	Yes (Note 6)	USB2_OC1#	GPI	None	•The Pull-down resistor is disabled after RSMRST# de-asserts
GPP_E11	20K PD (Note 4)	No	Yes (Note 6)	US2_OC2#	GPI	None	•The Pull-down resistor is disabled after RSMRST# de-asserts

Table 17-2. General Purpose I/O Signals (Sheet 7 of 9)

Name	Internal Pull-Up/ Pull-Down (Note 1)	De-Glitch (Note 2)		Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 3)	Default	NMI or SMI Capable	Notes
		Input	Output				
GPP_E12	20K PD (Note 4)	No	Yes (Note 6)	USB2_OC3#	GPI	None	•The Pull-down resistor is disabled after RSMRST# de-asserts
GPP_E13	None	No	Yes (Note 6)	DDPB_HPD0	GPI	NMI SMI	
GPP_E14	None	No	Yes (Note 6)	DDPC_HPD1	GPI	NMI SMI	
GPP_E15	None	No	Yes (Note 6)	DDPD_HPD2	GPI	NMI SMI	
GPP_E16	None	No	Yes (Note 6)	DDPE_HPD3	GPI	NMI SMI	
GPP_E17	None	No	Yes (Note 6)	EDP_HPD	GPI	None	
GPP_E18	None	Yes (Note 8)	Yes (Note 6)	DDPB_CTRLCLK	GPI	None	
GPP_E19	20K PD (Note 4)	Yes (Note 8)	No	DDPB_CTRLDATA	GPO	None	•Also used as a strap. •The Pull-down resistor is disabled after PCH_PWROK de-asserts •As GPO, the signal defaults to '0
GPP_E20	None	Yes (Note 8)	Yes (Note 6)	DDPC_CTRLCLK	GPI	None	
GPP_E21	20K PD (Note 4)	Yes (Note 8)	No	DDPC_CTRLDATA	GPO	None	•Also used as a strap. •The Pull-down resistor is disabled after PCH_PWROK de-asserts •As GPO, the signal defaults to '0
GPP_E22	None	Yes (Note 8)	Yes (Note 6)	None	GPI	None	
GPP_E23	20K PD (Note 4)	Yes (Note 8)	No	None	GPO	None	•Also used as a strap. •The Pull-down resistor is disabled after PCH_PWROK de-asserts •As GPO, the signal defaults to '0
Group F GPIO - Primary Power Well (1.8V Only)							
GPP_F0	None	No	Yes (Note 6)	I2S2_SCLK	GPI	None	
GPP_F1	None	No	Yes (Note 6)	I2S2_SFRM	GPI	None	
GPP_F2	None	No	Yes (Note 6)	I2S2_TXD	GPI	None	
GPP_F3	None	No	Yes (Note 6)	I2S2_RXD	GPI	None	
GPP_F4	None	Yes (Note 8)	Yes (Note 6)	I2C2_SDA	GPI	None	

Table 17-2. General Purpose I/O Signals (Sheet 8 of 9)

Name	Internal Pull-Up/ Pull-Down (Note 1)	De-Glitch (Note 2)		Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 3)	Default	NMI or SMI Capable	Notes
		Input	Output				
GPP_F5	None	Yes (Note 8)	Yes (Note 6)	I2C2_SCL	GPI	None	
GPP_F6	None	Yes (Note 8)	Yes (Note 6)	I2C3_SDA	GPI	None	
GPP_F7	None	Yes (Note 8)	Yes (Note 6)	I2C3_SCL	GPI	None	
GPP_F8	None	Yes (Note 8)	Yes (Note 6)	I2C4_SDA	GPI	None	
GPP_F9	None	Yes (Note 8)	Yes (Note 6)	I2C4_SCL	GPI	None	
GPP_F10	None	Yes (Note 8)	Yes (Note 6)	I2C5_SDA (1st)/ ISH_I2C2_SDA (2nd)	GPI	None	
GPP_F11	None	Yes (Note 8)	Yes (Note 6)	I2C5_SCL (1st)/ ISH_I2C2_SCL (2nd)	GPI	None	
GPP_F12	None	No	Yes (Note 6)	EMMC_CMD	GPI	None	
GPP_F13	None	No	Yes (Note 6)	EMMC_DATA0	GPI	None	
GPP_F14	None	No	Yes (Note 6)	EMMC_DATA1	GPI	None	
GPP_F15	None	No	Yes (Note 6)	EMMC_DATA2	GPI	None	
GPP_F16	None	No	Yes (Note 6)	EMMC_DATA3	GPI	None	
GPP_F17	None	No	Yes (Note 6)	EMMC_DATA4	GPI	None	
GPP_F18	None	No	Yes (Note 6)	EMMC_DATA5	GPI	None	
GPP_F19	None	No	Yes (Note 6)	EMMC_DATA6	GPI	None	
GPP_F20	None	No	Yes (Note 6)	EMMC_DATA7	GPI	None	
GPP_F21	None	No	Yes (Note 6)	EMMC_RCLK	GPI	None	
GPP_F22	None	No	Yes (Note 6)	EMMC_CLK	GPI	None	
GPP_F23	None	No	Yes (Note 6)	None	GPI	None	
Group G GPIO - Primary Power Well (1.8V or 3.3V)							
GPP_G0	None	No	Yes (Note 6)	SD_CMD	GPI	None	
GPP_G1	None	No	Yes (Note 6)	SD_DATA0	GPI	None	
GPP_G2	None	No	Yes (Note 6)	SD_DATA1	GPI	None	
GPP_G3	None	No	Yes (Note 6)	SD_DATA2	GPI	None	

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

Name	Internal Pull-Up/ Pull-Down (Note 1)	De-Glitch (Note 2)		Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 3)	Default	NMI or SMI Capable	Notes
		Input	Output				
GPP_G4	None	No	Yes (Note 6)	SD_DATA3	GPI	None	
GPP_G5	None	No	Yes (Note 6)	SD_CD#	GPI	None	
GPP_G6	None	No	Yes (Note 6)	SD_CLK	GPI	None	
GPP_G7	None	No	Yes (Note 6)	SD_WP	GPI	None	
GPIO In Deep Sleep Power Well (3.3V Only)							
GPD0	None	No	Yes (Note 6)	BATLOW#	BATLOW#	None	
GPD1	None	No	Yes (Note 6)	ACPRESENT	ACPRESENT	None	
GPD2	None	No	Yes (Note 6)	LAN_WAKE#	LAN_WAKE#	None	
GPD3	None	Yes (Note 8)	Yes (Note 6)	PWRBTN#	PWRBTN#	None	
GPD4	None	No	Yes (Note 6)	SLP_S3#	SLP_S3#	None	
GPD5	None	No	yes (Note 6)	SLP_S4#	SLP_S4#	None	
GPD6	None	No	Yes (Note 6)	SLP_A#	SLP_A#	None	
GPD7	None	No	Yes (Note 6)	Reserved Functionality	Reserved Functionality (Needs to be programmed for GPIO use)	None	The reserved functionality is an output and defaults to '1' after reset.
GPD8	None	No	Yes (Note 6)	SUSCLK	SUSCLK	None	
GPD9	None	No	Yes (Note 6)	SLP_WLAN#	SLP_WLAN#	None	
GPD10	None	No	Yes (Note 6)	SLP_S5#	SLP_S5#	None	
GPD11	None	No	Yes (Note 6)	LANPHYPC	LANPHYPC	None	

Notes:
1. All GPIOs have weak internal Pull-up or Pull-down resistors that can be configured by BIOS—these resistors are off by default. The Pull-up/Pull-down resistor shown in this column is always present by default.
2. De-glitch signal has a ~20 KOhm Pull-down during the pin power sequencing
3. When only one function is multiplexed on a GPIO, that function is considered the First Native Function. A native function (Native Function 1, Native Function 2, or Native Function 3) that is multiplexed on a GPIO can be selected via the PAD MODE register bit field in corresponding PAD_CFG_DW0 register. Refer to the register for more info.
4. The Pull-down resister value ranges from 14 KOhm - 26 KOhm with nominal value of 20 KOhm and will be disabled after RSMRST# or PCH_PWROK de-assertion (as indicated in the table.)
5. LPC mode and eSPI mode are determined by HW ESPI Enable Strap. See the pin strap section for more detail.
6. The signal is high-Z output with no glitch-free Pull-down resistor during the pin power sequencing
7. The signal is high-Z output with glitch-free Pull-down resistor (~20 KOhm) during the pin power sequencing
8. Input De-Glitch is only implemented on native functionality (not on GPIO functionality)
9. The signal is high-Z output with glitch-free Pull-up resistor (~20 KOhm) during the pin power sequencing".

17.5 Integrated Pull-Ups and Pull-Downs

All GPIOs have weak internal Pull-up/Pull-down resistors which are off by default. The internal PU/PD can be programmed (PU/PD/None) by BIOS after reset.

See Table 17-2 (Internal Pull-up/Pull-down) for info on which GPIO have integrated PU/PD enabled by default.

17.6 Functional Description

17.6.1 SMI#/SCI and NMI

SCI capability is available on all GPIOs, while NMI and SMI 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]

17.6.2 Blink/PWM Capability

The PCH provides blink/PWM capability on GPP_D[4:0].

To enable blink/PWM capability, the Pad Mode (in PAD_CFG_DW0_GPD_x register) of the corresponding GPIO needs to be set to 02h. See the Datasheet Volume 2 for more info on the register.

Software controls the blink/PWM by updating the PWM Control register and setting the SW update (SWUP) bit whenever a change in frequency or duty cycle of the PWM output signal is required. The new settings is applied at the start of the next output cycle and resets the SWUP bit.

The PWM output is controlled by two different settings:

- Frequency is controlled by the BASEUNIT setting of the PWM Control register. The BASEUNIT value is added to a 24 bit counter every clock cycle and the counter roll-over marks the start of a new cycle.
- Duty cycle is controlled by the On Time Divisor (ONTIMEDIV) setting (0 to 255). When the counter rolls-over it is reset and a new cycle starts with the output signal being 0, once the counter reaches the ONTIMEDIV value the output toggles to 1 and stays high until the counter rolls over.

The PWM module is clocked by PWM clock (32.768 KHz) and the output frequency can be estimated with the equation:

$$OutputFrequency = pwm_{clk} * \frac{base\_unit\_value}{256}$$

Note: With larger values of BASEUNIT the less resolution for controlling the duty cycle. For example, any BASEUNIT value greater than 128 will result in 16.384 KHz maximum frequency (with 32.768 KHz PWM clock) with no resolution for controlling the duty cycle. The maximum duty cycle resolution is 8 bits.

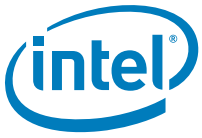


Table 17-3. PWM Output Frequencies Assuming 32.768 KHz

Output Frequency	Base Unit Value	CLK Cycle Count	
16.384 KHz	>128	1	No resolution
1.408 KHz	11	23	< 8 bit resolution
0.64 KHz	5	51	< 8 bit resolution
0.128 KHz	1	256	8 bit resolution
0.064 KHz	0.5	512	>8 bit resolution
0	0	0	Flat 0 output

17.6.2.1 PWM Programing Sequence

To ensure that there are no blips or other operational issues with PWM the following programming sequences must be performed in the order defined.

- Initial Enable or First Activation
  - Program the Base Unit and On Time Divisor values
  - Set the Software Update Bit
  - Enable the PWM Output by setting the PWM Enable bit
  - Repeat the above steps for the next PWM module
- Dynamic update while PWM is Enabled
  - Program the Base Unit and On Time Divisor values
  - Set the Software Update Bit
  - Repeat the above steps for the next PWM module

17.6.3 Triggering

PCH GPIOs have “sticky” bits on the input. Refer to the GPE1_GPI_STS register, GPI_IS, GPI_NMI_STS, and the ALT_GPI_SMI_STS register. As long as the signal goes active for at least 2 clock cycles, the PCH keeps the sticky status bit active. The active level (high or low) can be selected in the GP_INV register. This does not apply to GPI_NMI_STS residing in GPIO I/O space.

If the system is in an S0 or an S1 state, the GPI inputs are sampled at 12 MHz, so the signal only needs to be active for about 166.67 ns to be latched. In the S3 – S5 states, the GPI inputs are sampled at 32.768 KHz, and thus must be active for at least 61 microseconds to be latched.

GPIs that are in the Primary well are not capable of waking the system from deep sleep state where the Primary well is not powered

If the input signal is still active when the latch is cleared, it will again be set (another edge is not required). This makes these signal “level’ triggered inputs.

17.6.4 Sx GPIO Implementation Considerations

The PCH GPIO groups are either in the Primary well or Deep Sleep well, which is on during S0 and Sx (S3-S5) power states. In some cases, the GPIO signals may be connected to devices powered by the core well on the motherboard, or may have Pull-up resistors to the core well. In this case, potential leakage current may occur during Sx states when the core well is off. The PADRSTCFG register bit in PAD_CFG_DW0_GPP_x register for each GPIO can be used to isolate the signal if needed. Another option for the isolation is to utilize the GPIORXDIS, GPIOTXDIS, and TERM register bits. Refer to the register bit description for details.

The following table shows GPIO configurations with recommendation for Sx isolation, when the signals are connected to a core-well device or have Pull-ups to the core well.

GPIO Pin Configuration	Recommendation	Comment
Defaults to GPIO and used as GPI	**Option 1:** BIOS configures PADRSTCFG to select PLTRST#. **Option 2:** BIOS disables RX path via GPIORXDIS bit and disables any enabled Pull-up resistor via TERM bit prior to Sx entry. ***Note:*** BIOS needs to ensure GPIO registers are restored appropriately when resuming from Sx.	With option 1, when PLTRST# asserts upon Sx entry, GPIORXDIS register bit defaults to '1', which blocks its RX path. Other register bits in PAD_CFG_DW0_GPP_x and PAD_CFG_DW1_GPP_x will be also reset to default values.
Defaults to GPIO and used as GPO	**Option 1:** BIOS configures PADRSTCFG to select PLTRST#. **Option 2:** BIOS disables TX path via GPIOTXDIS bit and disables any enabled Pull-up resistor via TERM bit prior to Sx entry. ***Note:*** BIOS needs to ensure GPIO registers are restored appropriately when resuming from Sx.	With option 1, when PLTRST# upon Sx entry, asserts, GPIOTXDIS register bit defaults to '1', which blocks its TX path. Other register bits in PAD_CFG_DW0_GPP_x and PAD_CFG_DW1_GPP_x will be also reset to default values.
Defaults to GPIO and used as native function input and/or output	**Option 1:** BIOS configures PADRSTCFG to select PLTRST#. **Option 2:** BIOS disables TX and/or RX path via GPIOTXDIS and GPIORXDIS bit and disables any enabled Pull-up resistor via TERM bit prior to SX entry. ***Note:*** BIOS needs to ensure GPIO registers are restored appropriately when resuming from Sx.	With option 1, when PLTRST# asserts upon Sx entry, GPIORXDIS/GPIOTX register bit defaults to '1', which blocks its RX/TX path. Other register bits in PAD_CFG_DW0_GPP_x and PAD_CFG_DW1_GPP_x will be also reset to default values. The signals also revert back to GPIOs.
Defaults to native function	Keeps PADRSTCFG at default value	Some native signals may already be required to connect to suspend-well devices or Pull-ups, or drive '0' in Sx. For other cases, the PCH handles the isolation (such as PC controller).

17.6.5 GPIO Ownership

Any PCH GPIO can be owned either by the host or the Intel® ME. 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.

17.6.6 GPIO Pad Voltage Tolerance Configuration

Most GPIO pad voltage tolerance is determined by the power supplied to the associated power pin. For example, if VCCPGPPA is powered by 3.3V, all GPIO pads in GPIO group A are 3.3V tolerant. However, there is an exception to this rule. There are certain GPIO pads that can be configured to be 1.8V tolerant when the power supply is 3.3V, using the PAD_CFG_DW1 registers. These GPIO pads are:

- Group C: GPP_C[19:16]
- Group D: GPP_D[8:5] and GPP_D[14:13]

For more information, refer to the Datasheet Volume 2, PAD_CFG_DW1_x register for the associated GPIO pads.

§ §

18 Intel® Serial I/O Generic SPI (GSPI) Controllers

18.1 Acronyms

Acronyms	Description
GSPI	Generic Serial Peripheral Interface
LTR	Latency Tolerance Reporting

18.2 References

None

18.3 Overview

The PCH implements two generic SPI interfaces to support devices that use serial protocols for transferring data.

Each interface consists of 4 wires: a clock (CLK), a chip select (CS) and 2 data lines (MOSI and MISO).

18.4 Signal Description

Name	Type	Description
GSPI0_CS#/ GPP_B15	O	**Generic SPI 0 Chip Select**
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. See the pin strap section for more info.
GSPI1_CS#/ GPP_B19	O	**Generic SPI 1 Chip Select**
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. See the pin strap section for more info.

18.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
GSPI0_MOSI	Pull Down	9K - 50K	The integrated pull-down is disabled after PCH_PWROK assertion
GSPI1_MOSI	Pull Down	9K - 50K	The integrated pull-down is disabled after PCH_PWROK assertion
GSPI0_MISO	Pull Down	9K - 50K	
GSPI1_MISO	Pull Down	9K - 50K	

18.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
GSPI1_CS#, GSPI0_CS#	Primary	Undriven	Undriven	Undriven	Off
GSPI1_CLK, GSPI0_CLK	Primary	Undriven	Undriven	Undriven	Off
GSPI1_MISO, GSPI0_MISO	Primary	Undriven	Undriven	Undriven	Off
GSPI1_MOSI, GSPI0_MOSI	Primary	Internal P	Driven Low	Internal P	Off

18.7 Functional Description

18.7.1 Features

The GSPI interfaces support the following features:

- Full-duplex synchronous serial interface
- Support the Motorola's SPI protocol
- Operate in primary mode only
- 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)

Note: Secondary mode is not supported.

18.7.2 Controller Overview

The generic SPI controllers can only be set to operate as a primary.

The processor or DMA accesses data through the 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 bursts, 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 PCH 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 primary, 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.

18.7.3 DMA Controller

The GSPI controllers have an integrated DMA controller.

18.7.3.1 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. 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.

18.7.3.2 Channel Control

- The source transfer width and destination transfer width are programmed. 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.

18.7.4 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 the corresponding reset register to bring the controller from reset state into operational mode.

18.7.5 Power Management

18.7.5.1 Device Power Down Support

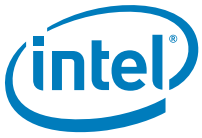
To power down peripherals connected to the PCH GSPI bus, the idle configured state of the I/O signals must be retained to avoid transitions on the bus that can affect the connected powered peripheral. Connected devices are allowed to remain in the D0 active or D2 low-power states when the bus is powered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

18.7.5.2 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. 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 represents 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.



18.7.6 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 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.

18.7.7 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.

§ §

19 Intel® Serial I/O Inter-Integrated Circuit (I2C) Controllers

19.1 Acronyms

Acronyms	Description
I2C	Inter-Integrated Circuit
PIO	Programmed Input/Output
SCL	Serial Clock Line
SDA	Serial Data Line

19.2 References

Specification	Location
The I2C Bus Specification, Version 5	www.nxp.com/documents/user_manual/UM10204.pdf?

19.3 Overview

The PCH implements six I2C controllers for six independent I2C 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 I2C host controllers and do not incorporate a DMA controller which is implemented for I2C0-I2C3. Therefore, I2C4 and I2C5 are restricted to operate in PIO mode only.

19.4 Signal Description

Name	Type	Description
I2C0_SDA/ GPP_C16	I/OD	**I2C Link 0 Serial Data Line** External Pull-up required.
I2C0_SCL/ GPP_C17	I/OD	**I2C Link 0 Serial Clock Line** External Pull-up required.
I2C1_SDA/ GPP_C18	I/OD	**I2C Link 1 Serial Data Line** External Pull-up required.
I2C1_SCL/ GPP_C19	I/OD	**I2C Link 1 Serial Clock Line** External Pull-up required.
I2C2_SDA/ GPP_F4	I/OD	**I2C Link 2 Serial Data Line** External Pull-up required.
I2C2_SCL/ GPP_F5	I/OD	**I2C Link 2 Serial Clock Line** External Pull-up required.

Name	Type	Description
I2C3_SDA/ GPP_F6	I/OD	**I2C Link 3 Serial Data Line** External Pull-up required.
I2C3_SCL/ GPP_F7	I/OD	**I2C Link 3 Serial Clock Line** External Pull-up required.
I2C4_SDA/ GPP_F8	I/OD	**I2C Link 4 Serial Data Line** External Pull-up required.
I2C4_SCL/ GPP_F9	I/OD	**I2C Link 4 Serial Clock Line** External Pull-up required.
I2C4B_SDA/ GPP_D13/ ISH_UART0_RXD/ SML0BDATA	I/OD	**2nd instance of the I2C Link 4 Data used for Comms Hub** External Pull-up required.
I2C4B_SCL/ GPP_D14/ ISH_UART0_TXD/ SML0BCLK	I/OD	**2nd instance of the I2C Link 4 Clock used for Comms Hub** External Pull-up required.
I2C5_SDA/ GPP_F10/ ISH_I2C2_SDA	I/OD	**I2C Link 5 Serial Data Line** External Pull-up required.
I2C5_SCL/ GPP_F11/ ISH_I2C2_SCL	I/OD	**I2C Link 5 Serial Clock Line** External Pull-up required.

19.5 Integrated Pull-Ups and Pull-Downs

None

19.6 I/O Signal Planes and States

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

19.7 Functional Description

19.7.1 Features

The I2C interfaces support the following features:

- Speed: standard mode (up to 100 Kb/s), fast mode (up to 400 Kb/s), and fast mode plus (up to 1 MB/s)
- 1.8V or 3.3V support (depending on the voltage supplied to the I2C signal group)
- Primary I2C 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 I2C used to share the I2C 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:

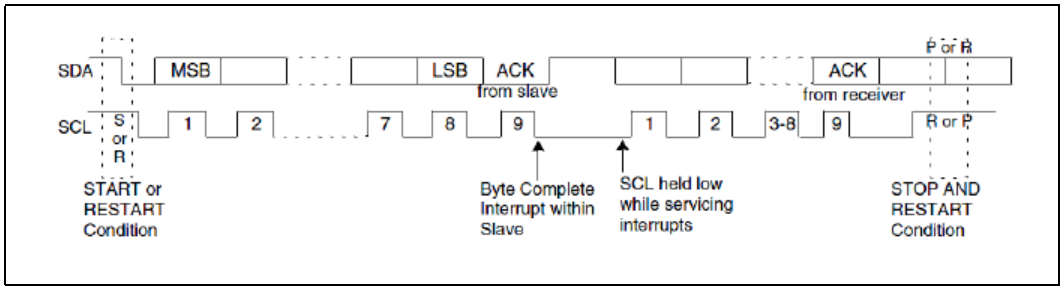
1. High-speed mode (up to 3.4 MB/s) is not supported.
2. The controllers must only be programmed to operate in primary mode only. I^2C secondary mode is not supported.
3. I^2C multi primaries are not supported.
4. Simultaneous configuration of Fast Mode and Fast Mode Plus is not supported.
5. I^2C General Call is not supported.

19.7.2 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 primary generates a START condition, signaling all devices on the bus to listen for data.
- The primary 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 primary 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 primary. 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 primary with a STOP condition. This frees the bus for the next primary to begin communications. When the bus is free, both data and clock lines are high.

Figure 19-1. Data Transfer on the I^2C Bus



19.7.2.1 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 primary, 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.

19.7.3 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.

19.7.3.1 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.

19.7.3.2 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 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.

19.7.4 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 off 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.

19.7.5 Power Management

19.7.5.1 Device Power Down Support

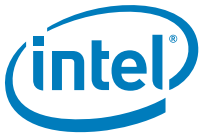
To power down peripherals connected to PCH I^2C bus, the idle configured state of the I/O signals is retained to avoid voltage transitions on the bus that can affect the connected powered peripheral. Connected devices are allowed to remain in the D0 active or D2 low-power states when I^2C bus is powered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

19.7.5.2 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 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.



19.7.6 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.

19.7.7 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.

19.7.8 Programmable SDA Hold Time

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

§ §

20 Gigabit Ethernet Controller

20.1 Acronyms

Acronyms	Description
GbE	Gigabit Ethernet

20.2 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	TBD

20.3 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. For details on the Intel® Ethernet Connection I219, refer to document (TBD). The Gigabit Ethernet Controller can operate at multiple speeds (10/100/1000 Mbps) and in either full duplex or half-duplex mode.

20.4 Signal Description

Table 20-1. GbE LAN Signals (Sheet 1 of 2)

Name	Type	Description
PCIE3_TXP **PCIE3_TXN** **PCIE4_TXP** **PCIE4_TXN** **PCIE5_TXP** **PCIE5_TXN** **PCIE9_TXP** **PCIE9_TXN** **PCIE10_TXP** **PCIE10_TXN**	O	Refer to Chapter 25 for details on the PCI Express transmit signals. ***Note:*** For PCH U/Y, the Intel® Ethernet Connection I219 can be connected to one of the following PCI Express ports 3, 4, 5, 9 and 10

Table 20-1. GbE LAN Signals (Sheet 2 of 2)

Name	Type	Description
PCIE3_RXP **PCIE3_RXN** **PCIE4_RXP** **PCIE4_RXN** **PCIE5_RXP** **PCIE5_RXN** **PCIE9_RXP** **PCIE9_RXN** **PCIE10_RXP** **PCIE10_RXN**	I	Refer to Chapter 25 for details on the PCI Express receive signals. ***Note:*** For PCH-U/Y the Intel® Ethernet Connection I219 can be connected to one of the following PCI Express ports 3, 4, 5, 9 and 10
SML0DATA/GPP_C4	I/OD	Refer to Chapter 29 for details on the SML0DATA signal. ***Note:*** The Intel® Ethernet Connection I219 connects to SML0DATA signal.
SML0CLK/GPP_C3	I/OD	Refer to Chapter 29 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. SLP_LAN# will always be de-asserted in S0 and anytime SLP_A# is de-asserted ***Note:*** If Gigabit Ethernet Controller is statically disabled via soft-strap or BIOS, SLP_LAN# will be driven low.
LAN_WAKE#/GPD2	I	**LAN WAKE:** LAN Wake Indicator from the GbE PHY. ***Note:*** Signal can instead be used as GPD2.

20.5 Integrated Pull-Ups and Pull-Downs

Table 20-2. 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	15k-40k	

20.6 I/O Signal Planes and States

Table 20-3. Power Plane and States for Output Signals (Sheet 1 of 2)

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

Table 20-3. Power Plane and States for Output Signals (Sheet 2 of 2)

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
Note: 1. Based on wake events and Intel ME state					

Table 20-4. 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/ Internal Pull-down[2]
Notes: 1. Configurable 2. Configurable based on PMC configuration bit. '1' (pin will be driven by platform in DeepSx) -> Undriven; '0' (pin will NOT be driven by platform in DeepSx) -> Internal Pull-down (15k-40k) enabled					

20.7 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).

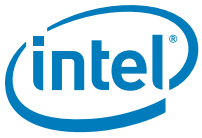
The Intel® Ethernet Connection I219 only runs at a speed of 1250 Mbps, which is 1/2 of the 2.5 GB/s PCI Express frequency. Each of the PCI Express* root ports in the PCH have the ability to run at the 1250-Mbps rate. There is no need to implement a mechanism to detect that the Platform LAN Device is 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 primary support for systems using more than 4 GB of physical memory
  - Programmable host memory receives 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 hardwares 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

20.7.1 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.

20.7.1.1 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.

20.7.1.2 Data Alignment

20.7.1.2.1 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.

20.7.1.2.2 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.

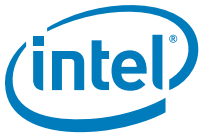
20.7.1.3 Configuration Request Retry Status

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.

20.7.2 Error Events and Error Reporting

20.7.2.1 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#.



20.7.2.2 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#.

20.7.3 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.

20.7.3.1 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 20-5.

Table 20-5. LAN Mode Support

Mode	System State	Interface Active	Connections
Normal 10/100/1000 Mbps	S0	PCI Express or SMLink0[1]	Intel® Ethernet Connection I219
Manageability and Remote Wake-up	Sx	SMLink0	Intel® Ethernet Connection I219
Note: [1]GbE operation is not supported in Sx state.			

20.7.4 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.

§ §

21 Interrupt Interface

21.1 Acronyms

Acronyms	Description
AEOI	Automatic End Of Interrupt
APIC	Advanced Programmable Interrupt Controller
HPET	High Precision Event Timer
PIC	Programmable Interrupt Controller

21.2 References

None

21.3 Overview

The interrupt controllers are used by the OS to dynamically route PCI interrupts to interrupt requests (IRQs).

21.4 Signal Description

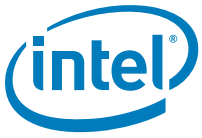
Name	Type	Description
SERIRQ/ GPP_A6	I/O	Serial Interrupt Request ***Note:*** An external Pull-up is required
PIRQA# / GPP_A7	I/OD	PCI Interrupt Request A ***Note:*** An external Pull-up is required

21.5 Integrated Pull-Ups and Pull-Downs

None

21.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
SERIRQ	Primary	Undriven	Undriven	Undriven	OFF
PIRQA#	Primary	Undriven	Undriven	Undriven	OFF



21.7 Functional Description

The PCH supports both APIC and PIC modes.

Interrupt sharing from the perspective of the Interrupt Controller that receives the Interrupts is limited to IRQ 0-23.

- Shareable interrupts require the Interrupt Controller to track the Assert/De-assert Sideband message from each interrupt source. The Interrupt Controller achieves this through Source ID decode of the message.
- Maintains backwards compatibility with the prior generations where only the lower 24 IRQs are available to support Interrupt Sharing.
- Interrupts are dedicated and not shareable from the perspective of the Interrupt Controller for IRQ 24-119. In other words, not more than 1 Interrupt Initiator is allowed to be assigned to the same IRQ# for IRQ 24-119. For example, GPIO (multi-cause Interrupt Initiator) and Intel® Serial I/O interfaces (I^2C, UART, GSPI) (multi-function Interrupt Initiator) should not both generate Assert/De-assert IRQn that maps to IRQ24.
- Possible multi-cause Interrupt Initiator that maps to IRQ24-119 are GPIO, eSPI, and so on.
- Possible multi-function Interrupt Initiators that maps to IRQ24-119 are HD Audio, I^2C/UART/GSPI (Intel Serial I/O Interfaces), Storage and Communication, ISH, and so on.

Interrupt Sharing Compliance Requirements for the Interrupt Initiator are as follows:

1. For multi-cause Initiators (Multiple Interrupt Causes from Single Source and Single SB Port ID, i.e. GPIO, eSPI): If more than 1 interrupt cause has to use the same IRQ#, it has to be aggregated or guaranteed through BIOS/SW to assign a unique IRQ per Interrupt Cause.
2. For multi-function devices (1 Interrupt Cause per Source but many Sources are behind Single SB Port ID, i.e., Intel® Serial I/O interfaces (I^2C, UART, GSPI)): Again if sharing is needed, the interrupts have to be aggregated or guaranteed through SW to ensure a unique IRQ is assigned per Interrupt Cause.
3. IPs that have 1:1 mapping to the IRQ# such as eSPI and LPC are not impacted by this requirement. For eSPI, it is expected that the EC devices aggregate the interrupts before these are communicated to eSPI.
4. Single-cause or Single-function device behind a unique SB Port ID is not subjected to this requirement.

Only level-triggered interrupts can be shared. PCI interrupts (PIRQs) are inherently shared on the board; these should, therefore, be programmed as level-triggered.

The following tables show the mapping of the various interrupts in Non-APIC and APIC modes.

Table 21-1. Interrupt Options - 8259 Mode (Sheet 1 of 2)

IRQ#	Pin	SERIRQ	PCI Message	Internal Modules
0	No	No	No	8254 Counter 0, HPET#0
1	No	Yes	No	Option for configurable sources including GPIO, eSPI and internal PCI/ACPI devices
2	No	No	No	8259 #2 cascade only

Table 21-1. Interrupt Options - 8259 Mode (Sheet 2 of 2)

IRQ#	Pin	SERIRQ	PCI Message	Internal Modules
3:7	PIRQA	Yes	Yes	Option for configurable sources including PIRQx, GPIO, eSPI and internal PCI/ACPI devices
8	No	No	No	RTC, HPET#1
9:10	PIRQA	Yes	Yes	Option for configurable sources including PIRQx, GPIO, eSPI, internal PCI/ACPI devices, SCI and TCO.
11	PIRQA	Yes	Yes	Option for configurable sources including PIRQx, GPIO, eSPI, internal ACPI devices, SCI, TCO, HPET #2
12	PIRQA	Yes	Yes	Option for configurable sources including PIRQx, GPIO, eSPI, internal ACPI devices, HPET#3
13	No	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI devices
14:15	PIRQA	Yes	Yes	Option for configurable sources including PIRQx, GPIO, eSPI and internal ACPI devices

Notes:
1. 8259 Interrupt Request Lines 0, 2 and 8 are non-shareable and dedicated. Only one interrupt source is allowed to use the Interrupt Request Line at any one time.
2. If an interrupt is used for PCI IRQ [A:H], SCI, or TCO, it should not be used for ISA-style interrupts (via SERIRQ).
3. In 8259 mode, PCI interrupts are mapped to IRQ3, 4, 5, 6, 7, 9, 10, 11, 12, 14, or 15. It can be programmed via 10.1.4 Interrupt Control Offset 60h-63h, 68h-6Bh.

Table 21-2. Interrupt Options - APIC Mode (Sheet 1 of 2)

IRQ#	Pin	SERIRQ	PCI Message	IRQ Sharable?	Internal Modules
0	No	No	No	No	Cascade from 8259 #1
1	No	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/ PCI devices
2	No	No	No	No	8254 Counter 0, HPET #0 (legacy mode)
3:7	No	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/ PCI devices
8	No	No	No	No	RTC, HPET #1 (legacy mode)
9:10	No	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/ PCI devices, SCI and TCO
11	No	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/ PCI devices, SCI, TCO, HPET #2
12	No	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/ PCI devices, HPET#3
13	No	No	No	Yes	Option for configurable sources including GPIO, eSPI and internal ACPI/PCI devices
14:15	No	Yes	No	Yes	Option for configurable sources including GPIO, eSPI and internal ACPI/PCI devices
16	PIRQA	PIRQA	Yes	Yes	Option for configurable sources including internal PIRQA, GPIO, eSPI and internal ACPI/PCI devices

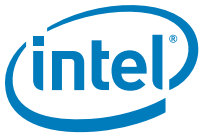


Table 21-2. Interrupt Options - APIC Mode (Sheet 2 of 2)

IRQ#	Pin	SERIRQ	PCI Message	IRQ Sharable?	Internal Modules
17:19	No	PIRQ[B-D]	Yes	Yes	Option for configurable sources including internal PIRQ[B-D], GPIO, eSPI and internal ACPI/PCI devices
20:23	No	No	No	Yes	Option for configurable sources including internal PIRQ[E-H], GPIO, eSPI, SCI, TCO, internal ACPI/PCI devices and HPET
24:119	No	No	No	No	Option for configurable sources including GPIO, eSPI and internal ACPI/PCI devices

Notes:
1. Interrupts 24 through 119 are dedicated and not shareable from the perspective of the Interrupt Controller. Not more than 1 Interrupt source is allowed to be assigned to the same IRQ#. For example, GPIO and Intel® Serial I/O interfaces (I^2C, UART, GSPI) should not generate Assert/Deassert_IRQn that maps to IRQ24. Although dedicated, Interrupts 24 through 119 can be configured to be level or edge-triggered.
2. If an interrupt is used for PCI IRQ [A:H], SCI, or TCO, it should not be used for ISA-style interrupts (via SERIRQ).
3. In APIC mode, the PCI interrupts [A:H] are directly mapped to IRQ[16:23].
4. When programming the polarity of internal interrupt sources on the APIC, interrupts 0 through 15, and 24 through 119 receive active-high internal interrupt sources; interrupts 16 through 23 receive active-low internal interrupt sources.
5. PIRQA is muxed with GPIO pins for assertion by external devices. Interrupt PIRQA will not be exposed if they are configured as GPIOs. When configured as GPIO pin, the internal PIRQA# is delivered internally to internal interrupt controller.
6. The internal ACPI/PCI devices refer to PCI/PCIe devices configured to the ACPI or PCI function mode. If in ACPI function mode, the device interrupt is map directly to one of the available IRQ. If in PCI function mode, the device interrupt is map to INT[A-D] and then to the IRQ before these devices issue the Interrupt Message using Assert/Deassert_IRQn.
7. PCI Message refers to the downstream Assert/Deassert_INT[A-D] messages forwarded from the processor complex.

The following signals are associated with the Interrupt Logic.

Table 21-3. Interrupt Logic Signals

Signal Name	C3	S1-D	S1-M	S3	S5
SERIRQ	Can be running	Tri-State (high)	Tri-State (high)	Off	Off
PIRQA#	Can go active	Tri-State (high)	Tri-State (high)	Off	Off

21.7.1 8259 Interrupt Controllers (PIC)

The ISA-compatible interrupt controller (PIC) incorporates the functionality of two 8259 interrupt controllers. The following table shows how the cores are connected.

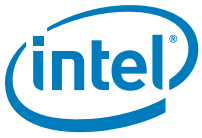
Table 21-4. Interrupt Controllers PIC

8259	8259 Input	Typical Interrupt Source	Connected Pin/Function
Primary	0	Internal	Internal Timer/Counter 0 output or Multimedia Timer #0
	1	Keyboard	IRQ1 via SERIRQ. Option for configurable sources including eSPI, GPIO, internal ACPI devices.
	2	Internal	Secondary Controller INTR output
	3	Serial Port A	IRQ3 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.
	4	Serial Port B	IRQ4 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.
	5	Parallel Port/Generic	IRQ5 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.
	6	Floppy Disk	IRQ6 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.
	7	Parallel Port/Generic	IRQ7 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.
Secondary	0	Real-time Clock	Inverted IRQ8# from internal RTC or Multimedia Timer #1
	1	Generic	IRQ9 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices, SCI, TCO.
	2	Generic	IRQ10 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices, SCI, TCO.
	3	Generic	IRQ11 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices, SCI, TCO or HPET #2.
	4	PS/2 Mouse	IRQ12 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices, SCI, TCO or HPET #3.
	5	Internal	IRQ13 from configurable sources including PIRQx, eSPI, GPIO, internal ACPI devices.
	6	Internal	IRQ14 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.
	7	Internal	IRQ15 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.

The secondary controller is cascaded onto the primary controller through primary controller interrupt input 2. This means there are only 15 possible interrupts for PCH PIC.

Interrupts can individually be programmed to be edge or level triggered, except for IRQ0, IRQ1, IRQ2 and IRQ8# which always default to edge.

Active-low interrupt sources, such as the PIRQ#s, are internally inverted before being sent to the PIC. In the following descriptions of the 8259s, the interrupt levels are in reference to the signals at the internal interface of the 8259s, after the required inversions have occurred. Therefore, the term “high” indicates “active”, which means “low” on an originating PIRQ#.



21.7.2 Interrupt Handling

21.7.2.1 Generating Interrupts

The PIC interrupt sequence involves three bits, from the IRR, ISR, and IMR, for each interrupt level. These bits are used to determine the interrupt vector returned, and status of any other pending interrupts. Table 21-5 defines the IRR, ISR, and IMR.

Table 21-5. Interrupt Status Registers

Bit	Description
IRR	**Interrupt Request Register.** This bit is set on a low to high transition of the interrupt line in edge mode, and by an active high level in level mode. This bit is set whether or not the interrupt is masked. However, a masked interrupt will not generate INTR.
ISR	**Interrupt Service Register.** This bit is set, and the corresponding IRR bit cleared, when an interrupt acknowledge cycle is seen, and the vector returned is for that interrupt.
IMR	**Interrupt Mask Register.** This bit determines whether an interrupt is masked. Masked interrupts will not generate INTR.

21.7.2.2 Acknowledging Interrupts

The processor generates an interrupt acknowledge cycle that is translated by the host bridge into a PCI Interrupt Acknowledge Cycle to the PCH. The PIC translates this command into two internal INTA# pulses expected by the 8259 cores. The PIC uses the first internal INTA# pulse to freeze the state of the interrupts for priority resolution. On the second INTA# pulse, the primary or secondary sends the interrupt vector to the processor with the acknowledged interrupt code. This code is based on Bits [7:3] of the corresponding ICW2 register, combined with three bits representing the interrupt within that controller.

Table 21-6. Content of Interrupt Vector Byte

Primary, Secondary Interrupt	Bits [7:3]	Bits [2:0]
IRQ7,15	ICW2[7:3]	111
IRQ6,14		110
IRQ5,13		101
IRQ4,12		100
IRQ3,11		011
IRQ2,10		010
IRQ1,9		001
IRQ0,8		000

21.7.2.3 Hardware/Software Interrupt Sequence

1. One or more of the Interrupt Request lines (IRQ) are raised high in edge mode, or seen high in level mode, setting the corresponding IRR bit.
2. The PIC sends INTR active to the processor if an asserted interrupt is not masked.
3. The processor acknowledges the INTR and responds with an interrupt acknowledge cycle. The cycle is translated into a PCI interrupt acknowledge cycle by the host bridge. This command is broadcast over PCI by the PCH.
4. Upon observing its own interrupt acknowledge cycle on PCI, the PCH converts it into the two cycles that the internal 8259 pair can respond to. Each cycle appears as an interrupt acknowledge pulse on the internal INTA# pin of the cascaded interrupt controllers.
5. Upon receiving the first internally generated INTA# pulse, the highest priority ISR bit is set and the corresponding IRR bit is reset. On the trailing edge of the first pulse, a secondary identification code is broadcast by the primary to the secondary on a private, internal three-bit wide bus. The secondary controller uses these bits to determine if it must respond with an interrupt vector during the second INTA# pulse.
6. Upon receiving the second internally generated INTA# pulse, the PIC returns the interrupt vector. If no interrupt request is present because the request was too short in duration, the PIC returns vector 7 from the primary controller.
7. This completes the interrupt cycle. In AEOI mode the ISR bit is reset at the end of the second INTA# pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

21.7.3 Initialization Command Words (ICWx)

Before operation can begin, each 8259 must be initialized. In the PCH, this is a four byte sequence. The four initialization command words are referred to by their acronyms: ICW1, ICW2, ICW3, and ICW4.

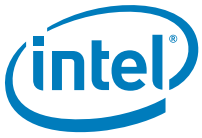
The base address for each 8259 initialization command word is a fixed location in the I/O memory space: 20h for the primary controller, and A0h for the secondary controller.

21.7.3.1 ICW1

An I/O write to the primary or secondary controller base address with data bit 4 equal to 1 is interpreted as a write to ICW1. Upon sensing this write, the PCH's PIC expects three more bytes writes to 21h for the primary controller, or A1h for the secondary controller, to complete the ICW sequence.

A write to ICW1 starts the initialization sequence during which the following automatically occur:

1. Following initialization, an interrupt request (IRQ) input must make a low-to-high transition to generate an interrupt.
2. The Interrupt Mask Register is cleared.
3. IRQ7 input is assigned priority 7.
4. The secondary mode address is set to 7.
5. Special mask mode is cleared and Status Read is set to IRR.



21.7.3.2 ICW2

The second write in the sequence (ICW2) is programmed to provide bits [7:3] of the interrupt vector that will be released during an interrupt acknowledge. A different base is selected for each interrupt controller.

21.7.3.3 ICW3

The third write in the sequence (ICW3) has a different meaning for each controller.

- For the primary controller, ICW3 is used to indicate which IRQ input line is used to cascade the secondary controller. Within the PCH, IRQ2 is used. Therefore, Bit 2 of ICW3 on the primary controller is set to a 1, and the other bits are set to 0s.
- For the slave controller, ICW3 is the secondary identification code used during an interrupt acknowledge cycle. On interrupt acknowledge cycles, the primary controller broadcasts a code to the secondary controller if the cascaded interrupt won arbitration on the primary controller. The secondary controller compares this identification code to the value stored in its ICW3, and if it matches, the secondary controller assumes responsibility for broadcasting the interrupt vector.

21.7.3.4 ICW4

The final write in the sequence (ICW4) must be programmed for both controllers. At the very least, Bit 0 must be set to a 1 to indicate that the controllers are operating in an Intel Architecture-based system.

21.7.4 Operation Command Words (OCW)

These command words reprogram the interrupt controller to operate in various interrupt modes.

- OCW1 masks and unmasks interrupt lines.
- OCW2 controls the rotation of interrupt priorities when in rotating priority mode, and controls the EOI function.
- OCW3 sets up ISR/IRR reads, enables/disables the special mask mode (SMM), and enables/disables polled interrupt mode.

21.7.5 Modes of Operation

21.7.5.1 Fully-Nested Mode

In this mode, interrupt requests are ordered in priority from 0 through 7, with 0 being the highest. When an interrupt is acknowledged, the highest priority request is determined and its vector placed on the bus. Additionally, the ISR for the interrupt is set. This ISR bit remains set until: the processor issues an EOI command immediately before returning from the service routine; or if in AEOI mode, on the trailing edge of the second INTA#. While the ISR bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels generate another interrupt. Interrupt priorities can be changed in the rotating priority mode.

21.7.5.2 Special Fully-Nested Mode

This mode is used in the case of a system where cascading is used, and the priority has to be conserved within each secondary. In this case, the special fully-nested mode is programmed to the primary controller. This mode is similar to the fully-nested mode with the following exceptions:

- When an interrupt request from a certain secondary is in service, this secondary is not locked out from the primary's priority logic and further interrupt requests from higher priority interrupts within the secondary are recognized by the primary and initiate interrupts to the processor. In the normal-nested mode, a secondary is masked out when its request is in service.
- When exiting the Interrupt Service Routine, software has to check whether the interrupt serviced was the only one from that secondary. This is done by sending a Non-Specific EOI command to the secondary and then reading its ISR. If it is 0, a Non-Specific EOI can also be sent to the primary.

21.7.5.3 Automatic Rotation Mode (Equal Priority Devices)

In some applications, there are a number of interrupting devices of equal priority. Automatic rotation mode provides for a sequential 8-way rotation. In this mode, a device receives the lowest priority after being serviced. In the worst case, a device requesting an interrupt has to wait until each of seven other devices are serviced at most once.

There are two ways to accomplish automatic rotation using OCW2: the Rotation on Non-Specific EOI Command (R=1, SL=0, EOI=1) and the rotate in automatic EOI mode which is set by (R=1, SL=0, EOI=0).

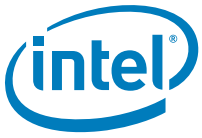
21.7.5.4 Specific Rotation Mode (Specific Priority)

Software can change interrupt priorities by programming the bottom priority. For example, if IRQ5 is programmed as the bottom priority device, then IRQ6 is the highest priority device. The Set Priority Command is issued in OCW2 to accomplish this, where: R=1, SL=1, and LO–L2 is the binary priority level code of the bottom priority device.

In this mode, internal status is updated by software control during OCW2. However, it is independent of the EOI command. Priority changes can be executed during an EOI command by using the Rotate on Specific EOI Command in OCW2 (R=1, SL=1, EOI=1 and LO–L2=IRQ level to receive bottom priority.

21.7.5.5 Poll Mode

Poll mode can be used to conserve space in the interrupt vector table. Multiple interrupts that can be serviced by one Interrupt Service Routine do not need separate vectors if the service routine uses the poll command. Poll mode can also be used to expand the number of interrupts. The polling Interrupt Service Routine can call the appropriate service routine, instead of providing the interrupt vectors in the vector table. In this mode, the INTR output is not used and the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by software using a Poll command.



The Poll command is issued by setting P=1 in OCW3. The PIC treats its next I/O read as an interrupt acknowledge, sets the appropriate ISR bit if there is a request, and reads the priority level. Interrupts are frozen from the OCW3 write to the I/O read. The byte returned during the I/O read contains a 1 in Bit 7 if there is an interrupt, and the binary code of the highest priority level in Bits 2:0.

21.7.5.6 Edge and Level Triggered Mode

In ISA systems this mode is programmed using Bit 3 in ICW1, which sets level or edge for the entire controller. In the PCH, this bit is disabled and a register for edge and level triggered mode selection, per interrupt input, is included. This is the Edge/Level control Registers ELCR1 and ELCR2.

If an ELCR bit is 0, an interrupt request will be recognized by a low-to-high transition on the corresponding IRQ input. The IRQ input can remain high without generating another interrupt. If an ELCR bit is 1, an interrupt request will be recognized by a high level on the corresponding IRQ input and there is no need for an edge detection. The interrupt request must be removed before the EOI command is issued to prevent a second interrupt from occurring.

In both the edge and level triggered modes, the IRQ inputs must remain active until after the falling edge of the first internal INTA#. If the IRQ input goes inactive before this time, a default IRQ7 vector is returned.

21.7.5.7 End Of Interrupt (EOI) Operations

An EOI can occur in one of two fashions: by a command word writes issued to the PIC before returning from a service routine, the EOI command; or automatically when AEOI bit in ICW4 is set to 1.

21.7.5.8 Normal End of Interrupt

In normal EOI, software writes an EOI command before leaving the Interrupt Service Routine to mark the interrupt as completed. There are two forms of EOI commands: Specific and Non-Specific. When a Non-Specific EOI command is issued, the PIC clears the highest ISR bit of those that are set to 1. Non-Specific EOI is the normal mode of operation of the PIC within the PCH, as the interrupt being serviced currently is the interrupt entered with the interrupt acknowledge. When the PIC is operated in modes that preserve the fully nested structure, software can determine which ISR bit to clear by issuing a Specific EOI. An ISR bit that is masked is not cleared by a Non-Specific EOI if the PIC is in the special mask mode. An EOI command must be issued for both the primary and secondary controller.

21.7.5.9 Automatic End of Interrupt Mode

In this mode, the PIC automatically performs a Non-Specific EOI operation at the trailing edge of the last interrupt acknowledge pulse. From a system standpoint, this mode should be used only when a nested multi-level interrupt structure is not required within a single PIC. The AEOI mode can only be used in the primary controller and not the secondary controller.

21.7.6 Masking Interrupts

21.7.6.1 Masking on an Individual Interrupt Request

Each interrupt request can be masked individually by the Interrupt Mask Register (IMR). This register is programmed through OCW1. Each bit in the IMR masks one interrupt channel. Masking IRQ2 on the primary controller masks all requests for service from the secondary controller.

21.7.6.2 Special Mask Mode

Some applications may require an Interrupt Service Routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The special mask mode enables all interrupts not masked by a bit set in the Mask Register. Normally, when an Interrupt Service Routine acknowledges an interrupt without issuing an EOI to clear the ISR bit, the interrupt controller inhibits all lower priority requests. In the special mask mode, any interrupts may be selectively enabled by loading the Mask Register with the appropriate pattern. The special Mask Mode is set by OCW3.SSMM and OCW3.SMM set, and cleared when OCW3.SSMM and OCW3.SMM are cleared.

21.7.7 Steering PCI Interrupts

The PCH can be programmed to allow PIRQ[A:D]# to be internally routed to interrupts 3-7, 9-12, 14 or 15, through the PARC, PBRC, PCRC, PDRC, PERC, PFRC, PGRC, and PHRC registers in the chipset configuration section. One or more PIRQx# lines can be routed to the same IRQx input.

The PIRQx# lines are defined as active low, level sensitive. When PIRQx# is routed to specified IRQ line, software must change the corresponding ELCR1 or ELCR2 register to level sensitive mode. The PCH will internally invert the PIRQx# line to send an active high level to the PIC. When a PCI interrupt is routed onto the PIC, the selected IRQ can no longer be used by an ISA device.

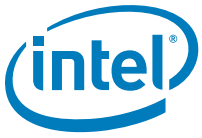
21.8 Advanced Programmable Interrupt Controller (APIC) (D31:F0)

In addition to the standard ISA-compatible PIC described in the previous section, the PCH incorporates the APIC. While the standard interrupt controller is intended for use in a uni-processor system, APIC can be used in either a uni-processor or multi-processor system.

21.8.1 Interrupt Handling

The I/O APIC handles interrupts very differently than the 8259. Briefly, these differences are:

- **Method of Interrupt Transmission.** The I/O APIC transmits interrupts through memory writes on the normal data path to the processor, and interrupts are handled without the need for the processor to run an interrupt acknowledge cycle.



- **Interrupt Priority.** The priority of interrupts in the I/O APIC is independent of the interrupt number. For example, interrupt 10 can be given a higher priority than interrupt 3.
- **More Interrupts.** The I/O APIC in the PCH supports a total of 24 interrupts.
- **Multiple Interrupt Controllers.** The I/O APIC architecture allows for multiple I/O APIC devices in the system with their own interrupt vectors.

21.8.2 Interrupt Mapping

The I/O APIC within the PCH supports 40 APIC interrupts. Each interrupt has its own unique vector assigned by software. The interrupt vectors are mapped as follows.

Table 21-7. APIC Interrupt Mapping[1]

IRQ #	Using SERIRQ	Direct from Pin	Using PCI Message	Internal Modules
0	No	No	No	Cascade from 8259 #1
1	Yes	No	Yes	
2	No	No	No	8254 Counter 0, HPET #0 (legacy mode)
3-7	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/PCI devices
8	No	No	No	RTC, HPET #1 (legacy mode)
9-10	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/PCI devices, SCI and TCO
11	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/PCI devices, SCI, TCO, HPET #2
12	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/PCI devices, HPET#3 (Note 3)
13	No	No	No	Option for configurable sources including GPIO, eSPI and internal ACPI/PCI devices
14-15	Yes	No	Yes	Option for configurable sources including GPIO, eSPI and internal ACPI/PCI devices
16	PIRQA#	PIRQA#[5]	Yes	Option for configurable sources including internal PIRQA, GPIO, eSPI and internal ACPI/PCI devices
17-19	PIRQ[B-D]#	No	Yes	Option for configurable sources including internal PIRQ[B-D], GPIO, eSPI and internal ACPI/PCI devices
20-23	No	No	No	Option for configurable sources including internal PIRQ[E-H], GPIO, eSPI, SCI, TCO, internal ACPI/PCI devices and HPET
24-119	No	No	No	Option for configurable sources including GPIO, eSPI and internal ACPI/PCI devices

Notes:
1. Interrupts 24 through 119 are dedicated and not shareable from the perspective of the Interrupt Controller. Not more than 1 Interrupt source is allowed to be assigned to the same IRQ#. For example, GPIO and Intel® Serial I/O interfaces (I^2C, UART, GSPI) should not generate Assert/Deassert_IRQn that maps to IRQ24. Although dedicated, Interrupts 24 through 119 can be configured to be level or edge-triggered.
2. If an interrupt is used for PCI IRQ [A:H], SCI, or TCO, it should not be used for ISA-style interrupts (using SERIRQ).
3. In APIC mode, the PCI interrupts [A:H] are directly mapped to IRQ[16:23].
4. When programming the polarity of internal interrupt sources on the APIC, interrupts 0 through 15, and 24 through 119 receive active-high internal interrupt sources; interrupts 16 through 23 receive active-low internal interrupt sources.

21.8.3 PCI/PCI Express* Message-Based Interrupts

When external devices through PCI/PCI Express wish to generate an interrupt, they will send the message defined in the *PCI Express* Base Specification,* Revision 2.0 for generating INTA# – INTD#. These will be translated internal assertions/de-assertions of INTA# – INTD#.

21.8.4 IOxAPIC Address Remapping

To support Intel Virtualization Technology (Intel VT), interrupt messages are required to go through similar address remapping as any other memory request. Address remapping allows for domain isolation for interrupts, so a device assigned in one domain is not allowed to generate an interrupt to another domain.

The address remapping is based on the Bus: Device: Function field associated with the requests. The internal APIC is required to initiate the interrupt message using a unique Bus: Device: Function.

The PCH allows BIOS to program the unique Bus: Device: Function address for the internal APIC. This address field does not change the APIC functionality and the APIC is not promoted as a stand-alone PCI device. See Device 31: Function 0 Offset 6Ch for additional information.

21.8.5 External Interrupt Controller Support

The PCH supports external APICs off of PCI Express ports but does not support APICs on the PCI bus. The EOI special cycle is only forwarded to PCI Express ports.

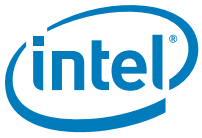
21.9 Serial Interrupt

The PCH supports a serial IRQ scheme. This allows a single signal to be used to report interrupt requests. The signal used to transmit this information is shared between the PCH and all participating peripherals. The signal line, SERIRQ, is synchronous to 24-MHz CLKOUT_LPC, and follows the sustained tri-state protocol that is used by all PCI signals. This means that if a device has driven SERIRQ low, it will first drive it high synchronous to PCI clock and release it the following PCI clock. The serial IRQ protocol defines this sustained tri-state signaling in the following fashion:

- **S – Sample Phase,** Signal driven low
- **R** – **Recovery Phase,** Signal driven high
- **T** – **Turn-around Phase,** Signal released

The PCH supports a message for 21 serial interrupts. These represent the 15 ISA interrupts (IRQ0–1, 3–15), the four PCI interrupts, and the control signals SMI# and IOCHK#. The serial IRQ protocol does not support the additional APIC interrupts (20–23).

Note: IRQ14 and IRQ15 are special interrupts and maybe used by the GPIO controller when it is running GPIO driver mode. When the GPIO controller operates in GPIO driver mode, IRQ14 and IRQ15 shall not be utilized by the SERIRQ stream nor mapped to other interrupt sources, and instead come from the GPIO controller. If the GPIO controller is entirely in ACPI mode, these interrupts can be mapped to other devices accordingly.



21.9.1 Start Frame

The serial IRQ protocol has two modes of operation which affect the start frame. These two modes are: Continuous, where the PCH is solely responsible for generating the start frame; and Quiet, where a serial IRQ peripheral is responsible for beginning the start frame.

The mode that must first be entered when enabling the serial IRQ protocol is continuous mode. In this mode, the PCH asserts the start frame. This start frame is 4, 6, or 8 PCI clocks wide based upon the Serial IRQ Control Register, bits 1:0 at 64h in D31:F0 configuration space. This is a polling mode.

When the serial IRQ stream enters quiet mode (signaled in the Stop Frame), the SERIRQ line remains inactive and pulled up between the Stop and Start Frame until a peripheral drives the SERIRQ signal low. The PCH senses the line low and continues to drive it low for the remainder of the Start Frame. Since the first PCI clock of the start frame was driven by the peripheral in this mode, the PCH drives the SERIRQ line low for 1 PCI clock less than in continuous mode. This mode of operation allows for a quiet, and therefore lower power, operation.Data Frames

Once the Start frame has been initiated, all of the SERIRQ peripherals must start counting frames based on the rising edge of SERIRQ. Each of the IRQ/DATA frames has exactly 3 phases of 1 clock each:

- **Sample Phase—**During this phase, the SERIRQ device drives SERIRQ low if the corresponding interrupt signal is low. If the corresponding interrupt is high, then the SERIRQ devices tri-state the SERIRQ signal. The SERIRQ line remains high due to Pull-up resistors (there is no internal Pull-up resistor on this signal, an external Pull-up resistor is required). A low level during the IRQ0–1 and IRQ2–15 frames indicates that an active-high ISA interrupt is not being requested, but a low level during the PCI INT[A:D], SMI#, and IOCHK# frame indicates that an active-low interrupt is being requested.
- **Recovery Phase—**During this phase, the device drives the SERIRQ line high if in the Sample Phase it was driven low. If it was not driven in the sample phase, it is tri-stated in this phase.
- **Turn-around Phase—**The device tri-states the SERIRQ line.

21.9.2 Stop Frame

After all data frames, a Stop Frame is driven by the PCH. The SERIRQ signal is driven low by the PCH for 2 or 3 PCI clocks. The number of clocks is determined by the SERIRQ configuration register. The number of clocks determines the next mode.

Table 21-8. Stop Frame Explanation

Stop Frame Width	Next Mode
2 PCI clocks	**Quiet Mode.** Any SERIRQ device may initiate a Start Frame
3 PCI clocks	**Continuous Mode.** Only the host (the PCH) may initiate a Start Frame

21.9.3 Specific Interrupts Not Supported Using SERIRQ

There are three interrupts seen through the serial stream that are not supported by the PCH. These interrupts are generated internally, and are not sharable with other devices within the system. These interrupts are:

- IRQ0. Heartbeat interrupt generated off of the internal 8254 counter 0.
- IRQ8#. RTC interrupt can only be generated internally.
- IRQ13. Reserved internally.

The PCH ignores the state of these interrupts in the serial stream, and does not adjust their level based on the level seen in the serial stream.Data Frame Format.

Table 21-9 shows the format of the data frames. For the PCI interrupts (A–D), the output from the PCH is AND'd with the PCI input signal. This way, the interrupt can be signaled using both the PCI interrupt input signal and using the SERIRQ signal (they are shared).

Table 21-9. Data Frame Format

Data Frame #	Interrupt	Clocks Past Start Frame	Comment
1	IRQ0	2	Ignored. IRQ0 can only be generated using the internal 8524
2	IRQ1	5	Before port 60h latch
3	SMI#	8	Causes SMI# if low. Will set the SERIRQ_SMI_STS bit.
4	IRQ3	11	
5	IRQ4	14	
6	IRQ5	17	
7	IRQ6	20	
8	IRQ7	23	
9	IRQ8	26	Ignored. IRQ8# can only be generated internally.
10	IRQ9	29	
11	IRQ10	32	
12	IRQ11	35	
13	IRQ12	38	Before port 60h latch
14	IRQ13	41	Ignored.
15	IRQ14	44	Not attached to GPIO logic
16	IRQ15	47	Not attached to GPIO logic
17	IOCHCK#	50	Same as ISA IOCHCK# going active
18	PCI INTA#	53	Drive PIRQA#
19	PCI INTB#	56	Drive PIRQB#
20	PCI INTC#	59	Drive PIRQC#
21	PCI INTD#	62	Drive PIRQD#

§ §

22 Integrated Sensor Hub (ISH)

22.1 Acronyms

Acronyms	Description
Intel® ME	Intel® Management Engine
I^2C	Inter-Integrated Circuit
IPC	Inter Process Communication
ISH	Integrated Sensor Hub
PMU	Power Management Unit
SRAM	Static Random Access Memory
UART	Universal Asynchronous Receiver/Transmitter

22.2 References

Specification	Location
I^2C Specification Version 5.0	http://www.nxp.com/documents/user_manual/UM10204.pdf

22.3 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® ME.

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, GPIO).
- An interface to main memory.
- Out of Band signals for clock and wake-up control.
- Inter Process Communications to the Host and Intel® ME.
- Part of the PCI tree on the host.

22.4 Signal Description

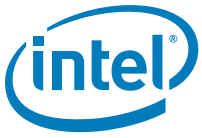
Name	Type	Description
ISH_I2C0_SDA/GPP_D5	I/OD	**I²C 0 Data**
ISH_I2C0_SCL/GPP_D6	I/OD	**I²C 0 Clk**
ISH_I2C1_SDA/GPP_D7	I/OD	**I²C 1 Data**
ISH_I2C1_SCL/GPP_D8	I/OD	**I²C 1 Clk**
ISH_I2C2_SDA /I2C5_SDA /GPP_F10	I/OD	**I²C 2 Data**
ISH_I2C2_SCL/I2C5_SCL **/**GPP_F11	I/OD	**I²C 2 Clk**
ISH_GP0/GPP_A18	I/O	**ISH GPIO 0**
ISH_GP1/GPP_A19	I/O	**ISH GPIO 1**
ISH_GP2/GPP_A20	I/O	**ISH GPIO 2**
ISH_GP3/GPP_A21	I/O	**ISH GPIO 3**
ISH_GP4/GPP_A22	I/O	**ISH GPIO 4**
ISH_GP5/GPP_A23	I/O	**ISH GPIO 5**
ISH_GP6/BM_BUSY#/GPP_A12	I/O	**ISH GPIO 6**
ISH_GP7/SD_PWR_EN# /GPP_A17	I/O	**ISH GPIO 7**
ISH_UART0_TXD/GPP_D14/SML0BCLK/I2C4B_SCL	O	**UART 0 Transmit Data**
ISH_UART0_RXD /GPP_D13/SML0BDATA/I2C2_SDA	I	**UART 0 Receive Data**
ISH_UART0_RTS#/GPP_D15	O	**UART 0 Request To Send**
ISH_UART0_CTS#/GPP_D16/SML0BALERT#	I	**UART 0 Clear to Send**
ISH_UART1_TXD/UART1_TXD/GPP_C13	O	**UART 1 Transmit Data**
ISH_UART1_RXD/UART1_RXD/GPP_C12	I	**UART 1 Receive Data**
ISH_UART1_RTS#/UART1_RTS#/GPP_C14	O	**UART 1 Request To Send**
ISH_UART1_CTS#/UART1_CTS#/GPP_C15	I	**UART 1 Clear to Send**

22.5 Integrated Pull-Ups and Pull-Downs

None

22.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	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



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

22.7 Functional Description

22.7.1 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.

22.7.2 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® ME 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.

22.7.3 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.

22.7.3.1 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 Config space. It is used only in ACPI mode (that is, when the PCI config space is hidden).

22.7.3.2 DMA Controller

The DMA controller supports up to 64-bit addressing.

22.7.3.3 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.

22.7.3.4 PCI Power Management

PME is not supported in ISH.

22.7.4 Power Domains and Management

22.7.4.1 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.

22.7.4.2 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.

22.7.5 ISH IPC

The ISH has IPC channels for communication with the Host Processor and Intel® ME. 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® ME). The supported initiator -> target flows using this mechanism are shown in the table below

Table 22-1. IPC Initiator -> Target flows

Initiator	Target
ISH	Host processor
Host processor	ISH
ISH	Intel® ME
Intel® ME	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.

22.7.6 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.

22.7.7 ISH I²C Controllers

The ISH supports two I²C controllers capable of operating at speeds up to 1 Mbps each. The I²C controllers are completely independent of each other: they do not share any pins, memory spaces, or interrupts.

The ISH's I²C host controllers share the same general specifications:

- Primary Mode Only (all peripherals must be secondary devices)
- Support for the following operating speeds:
  - Standard mode: 100 Kbps
  - Fast Mode: 400 Kbps
  - Fast Mode Plus: 1 Mbps
- Support for both 7-bit and 10-bit addressing formats on the I²C bus
- FIFO of 64 bytes with programmable watermarks/thresholds

22.7.8 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.

22.7.9 ISH GPIOs

The ISH support eight dedicated GPIOs.

22.8 Embedded Location (Comms Hub)

Embedded Location is a FW IP off-load function running on ISH 3.0 that has interfaces to the wireless communication ingredients (Wi-Fi, discrete GNSS and WWAN) on the platform. It enables background communication capabilities for platform location identification while the system is in S0ix mode and help optimize power consumption.

The various location identification elements on the platform are mentioned in the table below. Note that embedded location currently only works with Intel ingredients mentioned below and not with any other 3rd party connectivity devices.

Connectivity Ingredient	Ingredient Name	Embedded Location Usage
Wi-Fi	Snowfield Peak Wi-Fi	Indoor Location
Discrete GNSS	CG2000	Outdoor Location
WWAN	726x	Cell ID - Used for improved outdoor and indoor location identification

Connectivity Ingredient	Ingredient Name	Embedded Location Usage
Sensors	Sensors connected to ISH	Used to provide accurate platform location by taking into account the sensor data in conjunction with other connectivity ingredients like Wi-Fi, GNSS, and WWAN

§ §

23 Low Pin Count (LPC)

23.1 Acronyms

Acronyms	Description
LPC	Low Pin Count

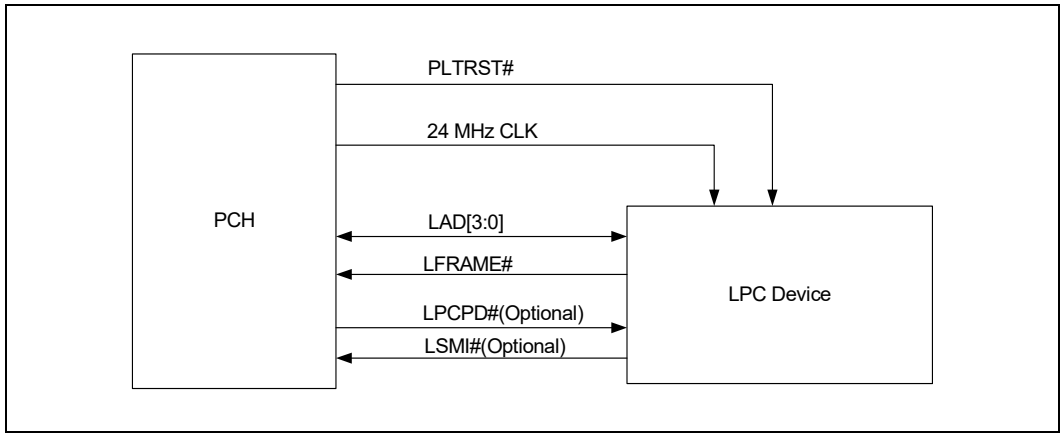
23.2 References

Specification	Location
Intel® Low Pin Count Interface Specification Revision 1.1	http://developer.intel.com/design/chipsets/industry/lpc.htm

23.3 Overview

The PCH implements an LPC interface as described in the *Low Pin Count Interface Specification, Revision 1.1*. The LPC interface to the PCH is shown in the following figure.

Figure 23-1. LPC Interface Diagram



The PCH supports all of the signals that are shown as optional, but peripherals are not required to do so.

LSMI# can be connected to any of the PCH's SMI capable GPIO signals.

LPCPD# can be connected to the PCH's SUS_STAT# if desired.

Note: The LPC bridge cannot be configured as a subtractive decode agent.

23.4 Signal Description

Name	Type	Description
LAD0/ ESPI_IO0/ GPP_A1	I/O	**LPC Multiplexed Command, Address, Data.** For LAD0, internal Pull-up is provided.
LAD1/ ESPI_IO1/ GPP_A2	I/O	**LPC Multiplexed Command, Address, Data.** For LAD1, internal Pull-up is provided.
LAD2/ ESPI_IO2/ GPP_A3	I/O	**LPC Multiplexed Command, Address, Data.** For LAD2, internal Pull-up is provided.
LAD3/ ESPI_IO3/ GPP_A4	I/O	**LPC Multiplexed Command, Address, Data.** For LAD3, internal Pull-up is provided.
LFRAME#/ ESPI_CS#/ GPP_A5	O	**LPC Frame:** LFRAME# indicates the start of an LPC cycle, or an abort.
RCIN#/ GPP_A0	I	**Keyboard Controller Reset Processor:** The keyboard controller can generate INIT# to the processor. This saves the external OR gate with the PCH's other sources of INIT#. When the PCH detects the assertion of this signal, INIT# is generated to the processor. ***Note:*** The PCH will ignore RCIN# assertion during transitions to the S3, S4, and S5 states.

23.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
LAD[3:0]	Pull-up	15 - 40 KΩ	

23.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
LAD[3:0]	Primary	Internal Pull-up	Internal Pull-up	Undriven	Off
LFRAME#	Primary	Driven High	Driven High	Driven Low	Off
RCIN#	Primary	Undriven	Undriven	Undriven	Off

23.7 Functional Description

The PCH LPC interface supports the *Low Pin Count Interface Specification, Revision 1.1*. The bus operates at 24-MHz clock frequency.

23.7.1 LPC Cycle Types

The PCH implements the cycle types shown in Table 23-1.

Table 23-1. LPC Cycle Types Supported (Sheet 1 of 2)

Cycle Type	Comment
Memory Read	1 byte only—(See Note 1 below)
Memory Write	1 byte only—(See Note 1 below)

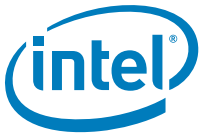


Table 23-1. LPC Cycle Types Supported (Sheet 2 of 2)

Cycle Type	Comment
I/O Read	1 byte only—The PCH breaks up 16-bit and 32-bit processor cycles into multiple 8-bit transfers.
I/O Write	1 byte only—The PCH breaks up 16-bit and 32-bit processor cycles into multiple 8-bit transfers.
Bus Primary Read	Can be 1, 2 or 4 bytes—(See Note 2 below)
Bus Primary Write	Can be 1, 2 or 4 bytes—(See Note 2 below)

Notes:

1. The PCH provides a single generic memory range (LGMR) for decoding memory cycles and forwarding them as LPC Memory cycles on the LPC bus. The LGMR memory decode range is 64 KB in size and can be defined as being anywhere in the 4-GB memory space. This range needs to be configured by BIOS during POST to provide the necessary memory resources. BIOS should advertise the LPC Generic Memory Range as Reserved to the OS in order to avoid resource conflict. For larger transfers, the PCH performs multiple 8-bit transfers. If the cycle is not claimed by any peripheral, it is subsequently aborted, and the PCH returns a value of all 1s to the processor. This is done to maintain compatibility with ISA memory cycles where pull-up resistors would keep the bus high if no device responds.
2. Primary Bus Read or Write cycles must be naturally aligned. For example, a 1-byte transfer can be to any address. However, the 2-byte transfer must be word-aligned (that is, with an address where A0=0). A DWord transfer must be DWord-aligned (that is, with an address where A1 and A0 are both 0)

23.7.2 Start Field Definition

Table 23-2. Start Field Bit Definitions

Bits[3:0] Encoding	Definition
0000	Start of cycle for a generic target
1111	Stop/Abort: End of a cycle for a target.

Note: All other encodings are RESERVED.

23.7.3 Cycle Type/Direction (CYCTYPE + DIR)

The PCH always drives Bit 0 of this field to 0. Table 23-3 shows the valid bit encodings.

Table 23-3. Cycle Type Bit Definitions

Bits[3:2]	Bit1	Definition
00	0	I/O Read
00	1	I/O Write
01	0	Memory Read
01	1	Memory Read
11	x	Reserved. If a peripheral performing a primary bus cycle generates this value, the PCH aborts the cycle.

Note: All other encodings are RESERVED.

23.7.4 Size

Bits[3:2] are reserved. The PCH always drives them to 00. Bits[1:0] are encoded as listed in Table 23-4.

Table 23-4. Transfer Size Bit Definition

Bits[1:0]	Size
00	8-bit transfer (1 byte)
01	16-bit transfer (2 bytes)
10	Reserved—The PCH never drives this combination.
11	32-bit transfer (4 bytes)

23.7.4.1 SYNC

Valid values for the SYNC field are shown in Table 23-5.

23.7.5 SYNC Timeout

Table 23-5. SYNC Bit Definition

Bits[3:0]	Indication
0000	**Ready:** SYNC achieved with no error.
0101	**Short Wait:** Part indicating wait-states. For primary bus cycles, the PCH does not use this encoding. Instead, the PCH uses the Long Wait encoding (see next encoding below).
0110	**Long Wait:** Part indicating wait-states, and many wait-states will be added. This encoding driven by the PCH for primary bus cycles, rather than the Short Wait (0101).
1010	**Error:** Sync achieved with error. This is generally used to replace the SERR# or IOCHK# signal on the PCI/ISA bus. It indicates that the data is to be transferred, but there is a serious error in this transfer.
Notes: 1. All other combinations are RESERVED. 2. If the LPC controller receives any SYNC returned from the device other than short (0101), long wait (0110), or ready (0000) when running a FWH cycle, indeterminate results may occur. A FWH device is not allowed to assert an Error SYNC.	

There are several error cases that can occur on the LPC interface. The PCH responds as defined in Section 4.2.1.9 of the *Low Pin Count Interface Specification*, Revision 1.1 to the stimuli described therein. There may be other peripheral failure conditions; however, these are not handled by the PCH.

23.7.6 SYNC Error Indication

The PCH responds as defined in Section 4.2.1.10 of the *Low Pin Count Interface Specification,* Revision 1.1.

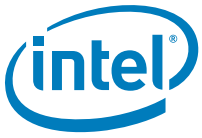
Upon recognizing the SYNC field indicating an error, the PCH treats this as a SERR by reporting this into the Device 31 Error Reporting Logic.

23.7.7 LFRAME# Usage

The PCH follows the usage of LFRAME# as defined in the *Low Pin Count Interface Specification,* Revision 1.1.

The PCH performs an abort for the following cases (possible failure cases):

- The PCH starts a Memory or I/O cycle, but no device drives a valid SYNC after four consecutive clocks.
- The PCH starts a Memory or I/O and the peripheral drives an invalid SYNC pattern.
- A peripheral drive an invalid value.



23.7.8 I/O Cycles

For I/O cycles targeting registers specified in the PCH's decode ranges, the PCH performs I/O cycles as defined in the *Low Pin Count Interface Specification,* Revision 1.1. These are 8-bit transfers. If the processor attempts a 16-bit or 32-bit transfer, the PCH breaks the cycle up into multiple 8-bit transfers to consecutive I/O addresses.

Note: If the cycle is not claimed by any peripheral (and subsequently aborted), the PCH returns a value of all 1s (FFh) to the processor. This is to maintain compatibility with ISA I/O cycles where Pull-up resistors would keep the bus high if no device responds.

23.7.9 LPC Power Management

23.7.9.1 LPCPD# Protocol

Same timings as SUS_STAT#. Upon driving SUS_STAT# low, the PCH drives LFRAME# low, and tri-states (or drives low) LAD[3:0].

Note: The *Low Pin Count Interface Specification,* Revision 1.1 defines the LPCPD# protocol where there is at least 30 µs from LPCPD# assertion to LRST# assertion. This specification explicitly states that this protocol only applies to entry/exit of low-power states which does not include asynchronous reset events. The PCH asserts both SUS_STAT# (connects to LPCPD#) and PLTRST# (connects to LRST#) at the same time during a global reset. This is not inconsistent with the LPC LPCPD# protocol.

23.7.10 Configuration and PCH Implications

23.7.10.1 LPC I/F Decoders

To allow the I/O cycles and memory mapped cycles to go to the LPC interface, the PCH includes several decoders. During configuration, the PCH must be programmed with the same decode ranges as the peripheral. The decoders are programmed using the D 31:F0 configuration space.

Note: The PCH cannot accept PCI write cycles from PCI-to-PCI bridges or devices with similar characteristics (specifically those with a "Retry Read" feature which is enabled) to an LPC device if there is an outstanding LPC read cycle towards the same PCI device or bridge. These cycles are not part of normal system operation, but may be encountered as part of platform validation testing using custom test fixtures.

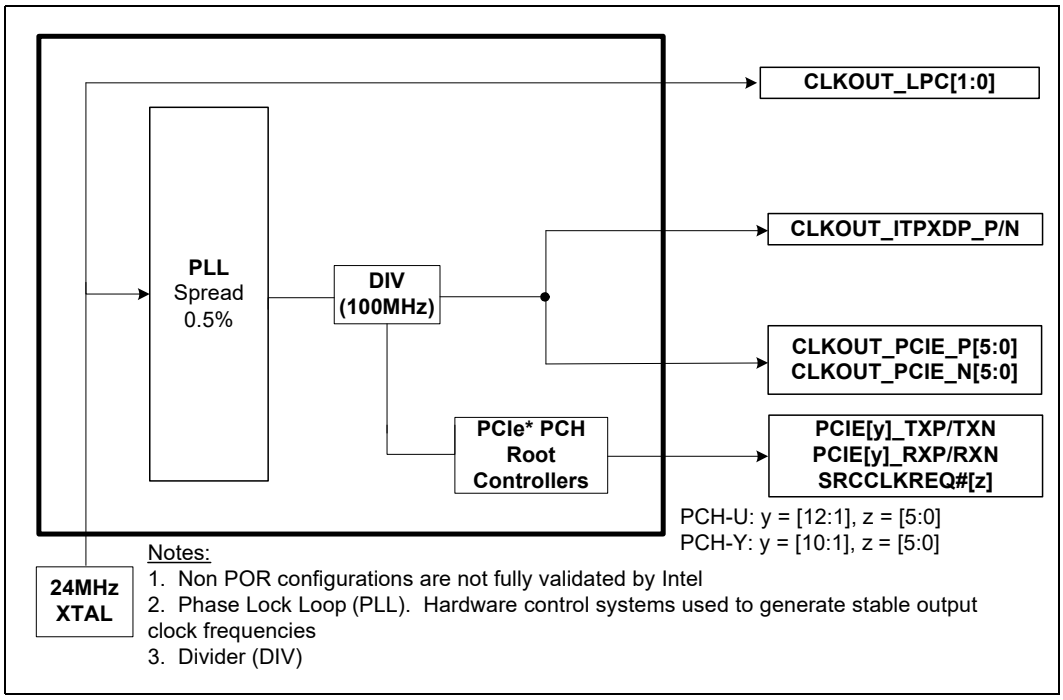
§ §

24 PCH and System Clocks

24.1 Overview

Platform Controller Hub (PCH) based platforms require several single-ended and differential clocks to synchronize signal operations and data propagations system wide between many interfaces and across multiple clock domains. The PCH generates and provides this complete system clocking solution through its Integrated Clock Controller (ICC).

Figure 24-1. PCH Internal Clock Diagram



24.2 Signal Descriptions

Name	Type	SSC Capable	Description
CLKOUT_ITPXDP_P CLKOUT_ITPXDP_N	O	Yes	**Differential ITP Debug Clock:** 100-MHz differential output to processor XDP/ITP connector on the platform
CLKOUT_PCIE_P[5:0] CLKOUT_PCIE_N[5:0]	O	Yes	**PCI Express* Clock Output:** 100 MHz PCIe* 3.0 specification-compliant differential output clocks to PCIe* devices
CLKOUT_LPC[1:0]	O	No	**Low Pin Count (LPC) Clock Outputs:** Single-Ended 24-MHz output to various single load connectors/devices
SRCCLKREQ#[5:0]	I/O	N/A	**Clock Request:** Clock request signals for PCIe* 100 MHz differential clocks
XTAL24_IN	I	N/A	**Crystal Input:** Input connection for 24-MHz crystal to PCH oscillator circuit

Name	Type	SSC Capable	Description
XTAL24_OUT	O	N/A	**Crystal Output:** Output connection for 24-MHz crystal to PCH oscillator circuit
XCLK_BIASREF	I/O	N/A	**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. N/A = Not Applicable
   — 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.

24.3 I/O Signal Planes and States

Table 24-1. I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
CLKOUT_ITPXDP_P CLKOUT_ITPXDP_N	Primary	Toggling	Toggling	Driven Low	OFF
CLKOUT_PCIE_P[5:0] CLKOUT_PCIE_N[5:0]	Primary	Toggling	Toggling	Driven Low	OFF
CLKOUT_LPC[1:0]	Primary	Toggling	Toggling	Driven Low	OFF
SRCCLKREQ#[5:0]	Primary	Un-driven	Un-driven	Un-driven	OFF
XTAL24_IN	Primary	Un-driven	Un-driven	Un-driven	OFF
XTAL24_OUT	Primary	Un-driven	Un-driven	Un-driven	OFF
XCLK_BIASREF	Primary	Un-driven	Un-driven	Un-driven	OFF

24.4 General Features

- The PCH Integrated Clock Controller (ICC) generates and supplies all the PCH reference clocks for internal needs and it provides the complete platform system clocking solution.
- All of the ICC PCH internal reference clocks and all of the single-ended and differential clock outputs are generated from an external 24-MHz crystal through the PCH XTAL24_IN pin, where the crystal accuracy is required to be less than ±30 ppm.
  Note: ppm stands for parts per million, and it indicates how much a crystal's frequency may deviate from the nominal value.
- CLKOUT_PCIE_P/CLKOUT_PCIE_N 100-MHz PCIe* 3.0 compliant differential output clocks support CLKREQ# based power management.
- CLKOUT_LPC[1:0] single-ended output clocks support CLKRUN# based power management, they require no external loop back clock for internal logic, and they only support a single load configuration.
- System Power Management support includes shutdown of all PCH ICC Phase Locked Loops (PLL), PCH ICC internal and external clocks, and includes the shutdown of the external 24-MHz crystal oscillator.

§ §

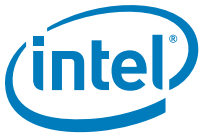
25 PCI Express* (PCIe*)

25.1 References

Specification	Location
PCI Express Base Specification*	http://www.pcisig.com/specifications
PCI Local Bus Specification	http://www.pcisig.com/specifications
PCI Power Management Specification	http://www.pcisig.com/specifications

25.2 Overview

- PCH-U supports up to 6 PCIe* Ports and 12 PCIe* Lanes, with transfer rates up to 8 GT/s (Gen3)
- PCH-Y supports up to 5 PCIe* Ports and 10 PCIe* Lanes, with transfer rates up to 8 GT/s (Gen3)
- PCI Express* Gen 1 and Gen 2 ExpressCard 1.0 module-based hot-plug support
- Dynamic Link Throttling
- Port 8xh Decode
- PCI Express* Gen 1 and Gen 2 Separate Reference Clock with Independent Spread Spectrum Clocking (SRIS) Support
- Latency Tolerance Reporting
- End-to-End PCI Express* Controller Lane Reversal
- Access Control Services
- Alternative Routing ID
- Autonomous Link Width Negotiation as a target
- Advanced Error Reporting
- PCI Express* Lane Polarity Inversion
- Configurable 128B or 256B Maximum Data Payload
- 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.
- Intel® Rapid Storage Technology (Intel® RST) for PCIe* Gen 1, Gen 2, and Gen 3 Storage Support
- PCI Express* Gen 1 and Gen 2 Receiver (RX) L0s Link Power Management State Support
- PCI Express* Gen 1, Gen 2, and Gen 3 External Graphics Support
- Single-Root I/O Virtualization (SR-IOV) Alternative Routing-ID Interpretation (ARI) and Access Control Services (ACS) feature support



25.3 Signal Description

PCH	Name	Type	Description
PCH-U	PCIE[12:1]_TXP PCIE[12:1]_TXN	O	**PCI Express* Differential Transmit Pairs 1 to 12** These are PCI Express* based outbound high-speed differential signals
	PCIE[12:1]_RXP PCIE[12:1]_RXN	I	**PCI Express* Differential Receive Pairs 1 to 12** These are PCI Express* based inbound high-speed differential signals
	PCIE_RCOMPP PCIE_RCOMPN	I	**Impedance Compensation Inputs**
PCH-Y	PCIE[10:1]_TXP PCIE[10:1]_TXN	O	**PCI Express* Differential Transmit Pairs 1 to 10** These are PCI Express* based outbound high-speed differential signals
	PCIE[10:1]_RXP PCIE[10:1]_RXN	I	**PCI Express* Differential Receive Pairs 1 to 10** These are PCI Express* based inbound high-speed differential signals
	PCIE_RCOMPP PCIE_RCOMPN	I	**Impedance Compensation Inputs**

25.4 I/O Signal Planes and States

Signal Name	Type	Power Plane	During Reset	Immediately After Reset	S3/S4/S5	Deep Sx
PCIE[12:1]_TXP PCIE[12:1]_TXN	O	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	Off
PCIE[12:1]_RXP PCIE[12:1]_RXN	I	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	Off
PCIE_RCOMPP PCIE_RCOMPN	I	Primary	Un-driven	Un-driven	Un-driven	Off

Note: PCIE1_RXP\RXN pins transition from un-driven to Internal Pull-down during Reset.

25.5 PCI Express* Port Support Feature Details

Table 25-1. PCI Express* Port Feature Details (Sheet 1 of 2)

PCH	Max. Device (Ports)	Max. Lanes	PCIe* Gen Type	Encoding	Transfer Rate (MT/s)	Theoretical Max. Bandwidth (GB/s)		
						x1	x2	x4
PCH-U	6	12	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
PCH-Y	5	10	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

Table 25-1. PCI Express* Port Feature Details (Sheet 2 of 2)

PCH	Max. Device (Ports)	Max. Lanes	PCIe* Gen Type	Encoding	Transfer Rate (MT/s)	Theoretical Max. Bandwidth (GB/s)		
						x1	x2	x4

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
3. See PCH PCIe* SkU specific feature breakdown details (Max. device support, Max. lane support, PCIe* Gen type) covered within the "Introduction" chapter

Table 25-2. PCI Express* Link Configurations Supported (Sheet 1 of 2)

PCH	PCIe* Link Config	Flexible HSIO Lanes											
		5	6	7	8	9	10	11	12	13	14	15	16
		PCIe* Controller 1				PCIe* Controller 2				PCIe* Controller 3			
		PCI Express* Lanes											
		1	2	3	4	5	6	7	8	9	10	11	12
PCH-U	1x4	P1				P5				P9			
	2x2	P1		P3		P5		P7		P9		P11	
	1x2 + 2x1	P1		P3	P4	P5		P7	P8	P9		P11	P12
	2x1 + 1x2	P4	P3	P1		P8	P7	P5		P12	P11	P9	
	4x1	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12
PCH-Y	1x4	P1				P5							
	2x2	P1		P3		P5		P7					
	1x2 + 2x1	P1		P3	P4	P5		P7	P8				
	2x1 + 1x2	P4	P3	P1		P8	P7	P5					
	4x1	P1	P2	P3	P4	P5	P6	P7	P8				
	1x2									P9			
	2x1									P9	P10		

Notes:
1. P# refers to a specific PCH PCI Express* Root Port #; for example P3 = 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* Port it connects to in the PCH (such as a x2 PCIe* Link connected to PCIe* Ports 3 and 4 would be called x2 PCIe* Port 3). This lowest number PCIe* Port in the PCIe* Link is known as the PCIe* Root Port.
3. The PCIe* Ports can be configured independently from one another but the Max. number of configured Devices (Ports) must not be exceeded
4. Unidentified Ports within a PCIe* Link Configuration are disabled but their physical lanes are used for the identified Port.
5. GbE can be mapped to PCIe* Root Ports 3, 4, 5, 9, or 10 within their respective PCIe* Link configurations. When GbE is enabled on a PCIe* Root Port, there can be at most up to a Max. of 5 Device (Ports) enabled.
6. PCH-U supports up to Two x4 or x2 re-mapped (Intel® Rapid Storage Technology) PCIe* SSD Gen 1/Gen 2/Gen 3 devices with a maximum of One re-mapped x4 or x2 device on PCIe* Controller #2 and a maximum of One re-mapped x4 or x2 device on PCIe* Controller #3
7. PCH-Y supports up to Two x4 or x2 re-mapped (Intel® Rapid Storage Technology) PCIe* SSD Gen 1/Gen 2/Gen 3 devices with a maximum of One re-mapped x4 or x2 device on PCIe* Controller #2 and a maximum of One re-mapped x2 device on PCIe* Controller #3.

Table 25-2. PCI Express* Link Configurations Supported (Sheet 2 of 2)

PCH	PCIe* Link Config	Flexible HSIO Lanes											
		5	6	7	8	9	10	11	12	13	14	15	16
		PCIe* Controller 1				PCIe* Controller 2				PCIe* Controller 3			
		PCI Express* Lanes											
		1	2	3	4	5	6	7	8	9	10	11	12

Notes: cont.

8. The PCH PCIe* Root Ports can be configured to map to any of the SRCCLKRQ# PCIe* clock request signals and the CLKOUT_SRC_P/N PCIe* differential signal pairs covered in the "PCH and System Clocks" Chapter
9. Reference and understand the PCIe* High Speed I/O Multiplexing details covered in the "Flexible I/O" Chapter
10. Supported Motherboard PCIe* Link Configuration Details
    - With PCIe* Controller Lane Reversal Disabled
      - PCH-U PCIe* Controller 1, 2, and 3 = 1x4, 2x2, 1x2+2x1, and 4x1
      - PCH-Y PCIe* Controller 1 and 2 = 1x4, 2x2, 1x2+2x1, and 4x1, PCIe* Controller 3 =1x2 and 2x1
    - With PCIe* Controller Lane Reversal Enabled
      - PCH-U PCIe* Controller 1, 2, and 3 = 1x4 and 2x1+1x2
      - PCH-Y PCIe* Controller 1 and 2 = 1x4 and 2x1+1x2
11. See PCH PCIe* SkU specific feature break down details (PCIe* Lane Mapping/Usage and Total Intel® RST for PCIe* Storage Devices) covered within the "Introduction" chapter

25.5.1 Intel® Rapid Storage Technology (Intel® RST) for PCIe* Storage

Intel® Rapid Storage Technology for PCIe* Storage provides an aggregation point for PCIe* and SATA storage devices enabling both of them to be controlled by the Intel® RST driver. This feature allows high performance PCIe* Gen 1/Gen 2/Gen 3 SSD devices to be used as stand-alone SSD devices. The Intel® RST driver re-maps the PCH architecture registers, that would be associated with a PCIe* SSD storage device, so that they can be easily accessible under the Integrated AHCI controller from a single driver software view point. This re-mapping is transparent to the PCIe* SSD storage devices and can only be accomplished with Intel® RST.

Note: The Intel® Rapid Storage Technology for PCIe* Storage is disabled by default upon PLTRST# de-assertion. During the discovery and initialization, BIOS enables this feature if functionality is required. Once enabled, this feature must not be disabled without going through the PLTRST#.

25.5.1.1 Supported Features Summary

- Concurrent support for 2 PCIe* SSD x2/x4 Gen 1, Gen 2 and Gen 3 interconnect devices
- 256-byte Max. payload size
- Early power shut down indication through the PME_Turn_Off message
- Only Intel® RST driver supported
- BIOS-assist during boot for the discovery and initialization sequence
- Hot-plug is not supported on PCIe* lanes enabled for the Intel® Rapid Storage Technology for PCIe* Storage

25.5.2 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 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 25-3 summarizes interrupt behavior for MSI and wire-modes. In the table "bits" refers to the hot-plug and PME interrupt bits.

Table 25-3. 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

25.5.3 Power Management

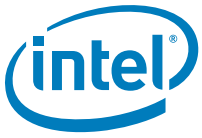
25.5.3.1 S3/S4/S5 Support

Software initiates the transition to S3/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 S3/S3/S5.

Prior to entering S3, software is required to put each device into $D3_{HOT}$. When a device is put into $D3_{HOT}$, it will initiate entries into a L1 link state by sending a PM_Enter_L1 DLLP. Under normal operating conditions when the root ports send the PME_Turn_Off message, the link will be in state L1. However, when the root port is instructed to send the PME_Turn_Off message, it will send it whether or not the link was in L1. Endpoints attached to the PCH can make no assumptions about the state of the link prior to receiving a PME_Turn_Off message.

25.5.3.2 Resuming from Suspended State

The root port contains enough circuitry in the suspend well to detect a wake event through the WAKE# signal and to wake the system. When WAKE# is detected asserted, an internal signal is sent to the power management controller of the PCH to cause the system to wake up. This internal message is not logged in any register, nor is an interrupt/GPE generated due to it.



25.5.3.3 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. See Section 25.5.3.4 for SMI/SCI generation.

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.

25.5.3.4 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.

When operating at PCIe* 8Gb/s, Link Equalization Request can also be routed to generate SCI or SMI. The intention8 for the SCI/SMI is to invoke the proprietary software to diagnose the reason behind the Link Equalization Request interrupt and take the proper link recovery path, which may include software re-performing link equalization. Root Ports do not support the hardware mechanism to service the Link Equalization Request from the device.

25.5.3.5 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).

25.5.4 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.

25.5.5 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.

25.5.6 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.

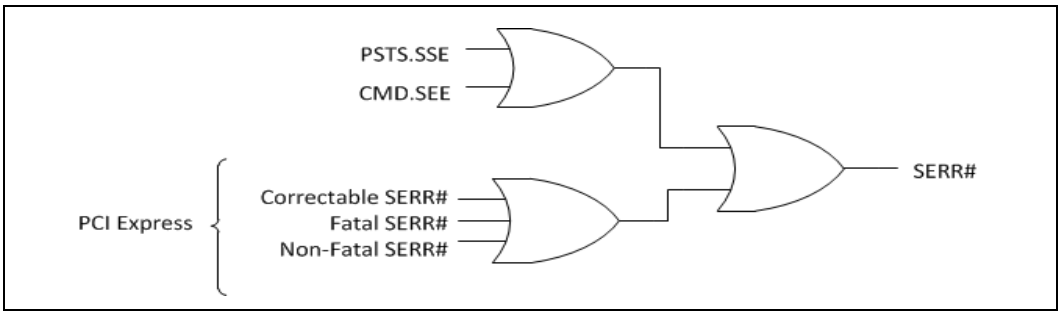
Soft straps are needed as a method to configure the port statically to operate in this mode. This mode is only enabled if the SSD connector is present on the motherboard, where the SSD connector does not include the reference clock. No change is being made to PCIe* add-in card form factors and solutions.

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.

25.5.7 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 25-1. Generation of SERR# to Platform



25.5.8 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

25.5.8.1 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.

25.5.8.2 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.

25.5.9 PCI Express* Lane Polarity Inversion

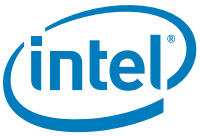
The PCI Express* Base Specification requires polarity inversion to be supported independently by all receivers across a Link—each differential pair within each Lane of a PCIe* Link handles its own polarity inversion. Polarity inversion is applied, as needed, during the initial training sequence of a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

25.5.10 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 and 2x1+1x2

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



26 Power Management

26.1 Acronyms

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

26.2 References

Specification	Location
Advanced Configuration and Power Interface, Version 4.0a (ACPI)	http://www.acpi.info/spec.htm

26.3 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.

26.4 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 ME 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 S3–S5 state. This signal can also be enabled to cause an SMI# when asserted. For Mobile package, this signal is multiplexed with GPD_0. This signal must be tied high to the VCCDSW_3p3, which will be tied to VCCPRIM_3p3 on Deep Sx disabled platforms. ***Note:*** Require external Pull-up to VCCDSW_3p3.
BM_BUSY# / GPP_A12 /ISH_GP6/ SX_EXIT_HOLDOFF#	I	**Primary Bus Busy:** Generic primary bus activity indication driven into the PCH. Can be configured to set the PM1_STS.BM_STS bit. Can also be configured to assert indications transmitted from the PCH to the processor using the PMSYNCH pin.

Name	Type	Description
CORE_VID0/GPP_B0	O	**PCH Core VID Bit 0**: May connect to external VRs and used to communicate the supported VCCPRIM_CORE voltage. ***Note:*** This pin will only be driven high ('1') in native mode, to reflect a VCCPRIM_CORE supported voltage of 1.0 V.
CORE_VID1/ GPP_B1	O	**PCH Core VID Bit 1:**May connect to external VRs and used to communicate the supported VCCPRIM_CORE voltage. ***Note:*** This pin will only be driven high ('1') in native mode, to reflect a VCCPRIM_CORE supported voltage of 1.0 V.
EXT_PWR_GATE#/ GPP_B11	O	External Power Gate for MPHY and SRAMs. ***Note:*** External MPHY/SRAM power gating has been de-featured on PCH-U/Y platforms. This pin, in native mode, will never be driven low.
DRAM_RESET#	OD O	**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. This input is tied together with RSMRST# on platforms that do not support Deep Sx. ***Note:*** This signal is in the RTC well.
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# during power-up and when S/W initiates a hard reset sequence through the Reset Control register (I/O port CF9h). The PCH drives PLTRST# active a minimum of 1 ms when initiated through the Reset Control register (I/O port CF9h). ***Note:*** PCI/PCIe* specification requires that the power rails associated with PCI/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.
PME#/GPP_A11	I/OD	**Power Management Event**: Driven by devices to wake the system or issue SCI.
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, 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 Sx.
RSMRST#	I	**Resume Well Reset:** This signal is used for resetting the resume power plane logic. This signal must be asserted for at least t201 after the suspend power wells are valid. When de-asserted, this signal is an indication that the suspend power wells are stable.
SLP_A#/GPD6	O	**SLP_A#:** Used to control power to the active sleep well (ASW) of the Platform. ***Note:*** There is no corresponding APWROK signal input to the PCH, but the PCH does have an internally generated version of APWROK that is timed from SLP_A#.

Name	Type	Description
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.
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. The selection between native and GPIO mode is based on a soft strap. The soft strap default is '0', slp_wlan# mode. Set soft strap to '1' to use the GPIO mode.
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. If Deep Sx is not supported, then this pin can be left unconnected. ***Note:*** This pin is in the DSW power well.
SUSACK#/GPP_A15	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_A13	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 ME 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_A13	O	**SUSPWRDNACK:** Active high. Asserted by the PCH on behalf of the Intel ME 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_A12 / BM_BUSY#/ISH_GP6	I	**Sx Exit Holdoff Delay**: Delay exit from Sx state after SLP_A# is de-asserted. See Section 26.7.9.5 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

Name	Type	Description
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.
SYS_RESET#	I	**System Reset**: This pin forces an internal reset after being de-bounced. The PCH will reset immediately if the SMBus is idle; otherwise, it will wait up to 25 ms ±2 ms for the SMBus to idle before forcing a reset on the system.
VRALERT#/GPP_B2	I	**VR Alert:** ICC Max. throttling indicator for the PCH voltage regulators.
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:*** External Pull-up required.
CLKRUN#/GPP_A8	I/OD	**LPC Clock Run:** Used to control CLKOUT_LPC[1:0]. Connects to peripherals that need to request clock restart or prevention of clock stopping.
SUS_STAT#/ ESPI_RESET#/ GPP_A14	O	**LPC Mode - Suspend Status:** This signal is asserted by the PCH to indicate that the system will be entering a low-power state soon. This can be monitored by devices with memory that need to switch from normal refresh to suspend refresh mode. It can also be used by other peripherals as an indication that they should isolate their outputs that may be going to powered-off planes. ***Note:*** In eSPI Mode, this signal functions as ESPI Reset#. Reset signal from PCH to secondary eSPI.

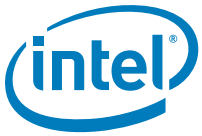
26.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
ACPRESENT/GPD1	Pull-down	15 KΩ – 40 KΩ	1
LAN_WAKE#/GPD2	Pull-down	15 KΩ – 40 KΩ	1
PWRBTN#/GPD3	Pull-up	15 KΩ – 40 KΩ	
PME#/GPP_A11	Pull-up	15 KΩ – 40 KΩ	
SUSACK#/GPP_A15	Pull-up	15 KΩ – 40 KΩ	
WAKE#	Pull-down	15 KΩ – 40 KΩ	1

Notes:
1. Pull-down is configurable and can be enabled in Deep Sx state; refer to DSX_CFG register (RCBA+3334h) for more details.

26.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
BATLOW#	DSW	Undriven	Undriven	Undriven	Undriven
BMBUSY#[15]	Primary	Undriven	Undriven	Undriven	Off
RSMRST#	RTC	Undriven	Undriven	Undriven	Undriven
PCH_PWROK	RTC	Undriven	Undriven	Undriven	Undriven
SYS_PWROK[13]	Primary	Undriven	Undriven	Undriven	Off
DSW_PWROK	RTC	Undriven	Undriven	Undriven	Undriven
DRAM_RESET#[14]	DSW	Undriven	Undriven	Undriven	Undriven



Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
VR_ALERT#[15]	Primary	Undriven	Undriven	Undriven	Off
SLP_S0#[1,6]	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]
SUSWARN#/ SUSPWRDNACK[6,10, 16]	Primary	Driven Low	Driven Low	Driven Low[5]	Off
SUSACK#[15]	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up	Off
CORE_VID0[11,16]	Primary	Driven High	Driven High	Driven High	Off
CORE_VID1[11,16]	Primary	Driven High	Driven High	Driven High	Off
ACPRESENT[6,10,15]	DSW	Undriven / Driven Low[4]	Undriven	Undriven	Undriven/ Driven Low[8]
WAKE#[13]	DSW	Undriven	Undriven	Undriven	Undriven/ Driven Low[8]
LAN_WAKE#[15]	DSW	Undriven	Undriven	Undriven	Undriven/ Driven Low[8]
LANPHYPC[10,16]	DSW	Driven Low	Driven Low	Driven Low	Driven Low
PME#[15]	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up	Off
PWRBTN#[15]	DSW	Internal Pull-up	Internal Pull-up	Internal Pull-up	Internal Pull-up
SYS_RESET#[13]	Primary	Undriven	Undriven	Undriven	Off
PLTRST#[16]	Primary	Driven Low	Driven High	Driven Low	Off
EXT_PWR_GATE#[16]	Primary	Driven High	Driven High	Driven High	Off
SX_EXIT_HOLDOFF#[15]	Primary	Z	Z	Z	Off

Signal Name	Power Plane	During Reset	Immediately after Reset	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 ME 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 ME state.
8. Pull-down is configurable and can be enabled in Deep Sx state; refer to DSX_CFG register (RCBA+3334h) 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 resistor is default off but configurable (pu/pd/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 1.0 V.
12. Pin state is a function of whether the platform is configured to have Intel ME on or off in Sx.
13. Output High-Z, not glitch free with ~ kΩ Pull-down during respective power sequencing.
14. Output High-Z, glitch free with ~ kΩ Pull-down during respective power sequencing
15. Output High-Z, not glitch free with ~20 kΩ Pull-down during respective power sequencing.
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.

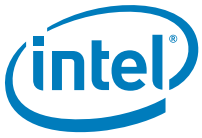
26.7 Functional Description

26.7.1 Features

- Support for *Advanced Configuration and Power Interface, Version 4.0a (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

26.7.2 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.

26.7.2.1 PCH S0 Low Power State Definition

A high level description of the global PCH low power states are described in table 1. 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 26-1. 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[3]
LP2	Pervasively Idle S0 and Root PLLs are off	C6 or deeper	All actions from LP1 + • Gen 2 PLL/BCLK PLL shutdown[4]
LP3	Idle Floor	C10	All actions from LP2 + • 24 MHz 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 shut down dynamically. This includes PCIe Gen3, SATA, USB2 and MIPI.
2. Internal Power Gating of PCH controllers – Each host controller (ie XHCI, AHCI), PCIe root port or embedded subsystem (ISH, CSME, Audio) when becoming sufficiently idle can autonomously power gate its core digital logic and local memory arrays.
3. Internal HSIO Per Lane Power gating – New on PCH-U/Y – The PCH HSIO lanes (PCIe/SATA/USB3/USB2) support dynamic power gating on a per lane basis, meaning when the respective lane enters the appropriate link power state it can be power gated irrespective of the remaining potentially active lanes.
4. Gen 2 PLL/BCLK PLL shutdown – The PCH IOSF Primary clocks and PLL can be gated when all host controllers and I/O interfaces are sufficiently idle and in the appropriate link power states. Additionally, since the same PLL (depending on configuration) also supplies the BCLK to the CPU, BCLK PLL may also be shut down if the minimum Pkg C-state level is met as well.

26.7.2.2 24 MHz Crystal Shutdown

When the CPU and system are in a power management state that can tolerate gating the 24 MHz crystal clock, this circuit can be powered down. This occurs when the processor enters C10 state, the PCH is in LP3 and all other consumers of the 24 MHz XTAL de-assert their clock request.

26.7.2.3 SLP_S0#

SLP_S0# is the indication to the system to enter the deterministic idle state (S0i3). This is a PCH hardware controlled output pin. This signal is defined as active low which means a 0V indicates the deterministic idle state. Additional power saving steps such as VPCLVM may happen during this state.

Note: VPCLVM is not supported on the 6th Generation Intel® Core™ processor U/Y platforms.

26.7.2.4 VCCPRIM_CORE Low Voltage Mode (VPCLVM)

When SLP_S0# asserts and the PCH enters the deterministic idle state, the power supplied to the VCCPRIM_CORE rail can transition to a lower 0.7 V level to further reduce the PCH idle power. PMIC or discrete VR solutions that support this low voltage mode would use the SLP_S0# input assertion as indication of entry into VPCLVM and de-assertion as an indication to exit VPCLVM.

Notes:

1. The VCCPRIM_CORE voltage level during VPMLVM is lower than the active 1.0 V voltage level.
2. VPCLVM is not supported on the 6th Generation Intel® Core™ processor U/Y platforms.

26.7.3 PCH and System Power States

Table 26-2 shows the power states defined for PCH-based platforms. The state names generally match the corresponding ACPI states.

Table 26-2. General Power States for Systems Using the PCH

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.
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.
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 S3 state, then the resume path will place system back into S3. 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 GEN_PMCON_3 register (D31:F0, offset A4). Refer to Table 26-8 for more details.

Table 26-3 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 26-3.

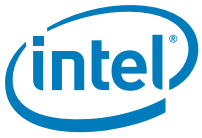


Table 26-3. State Transition Rules for the PCH

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] • Conditions met as described in Section 26.7.8.6.1 and Section 26.7.8.6.2 • Mechanical Off/Power Failure	• G0/S0/C0[2] • G2/S5 • Deep Sx • G3
G1/S4	• Any Enabled Wake Event • Power Button Override[3,5] • Conditions met as described in Section 26.7.8.6.1 and Section 26.7.8.6.2 • Mechanical Off/Power Failure	• G0/S0/C0[2] • G2/S5 • Deep Sx • G3
G2/S5	• Any Enabled Wake Event • Conditions met as described in Section 26.7.8.6.1 and Section 26.7.8.6.2 • Mechanical Off/Power Failure	• G0/S0/C0[2] • Deep Sx • G3
G2/Deep Sx	• Any Enabled Wake Event • ACPRESENT Assertion • Mechanical Off/Power Failure	• G0/S0/C0[2] • G1/S3, G1/S4 or G2/S5 (see Section 26.7.8.6.2) • G3
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 Sx is enabled and conditions are met per Section 26.7.8.6, the system will transition to Deep Sx.

26.7.4 System Power Planes

The system has several independent power planes, as described in Table 26-4.

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

Table 26-4. System Power Plane

Plane	Controlled By	Description
Processor	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, LPC I/F, and 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 id 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® ME	SLP_A#	SLP_A# signal is asserted when the Intel ME platform goes to M-Off. Depending on the platform, this pin may be used to control power to various devices that are part of the Intel ME sub-system in the platform.
LAN	SLP_LAN#	This signal is asserted in Sx/M-Off when both host and Intel ME WoL are not supported. This signal can be used 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.
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.

26.7.5 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.

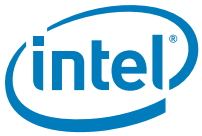


Table 26-5 shows which events can cause an SMI and SCI.

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 26-5. Causes of SMI and SCI (Sheet 1 of 2)

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	PRBTNOR_STS
RTC Alarm	Yes	Yes	RTC_EN=1	RTC_STS
ACPI Timer overflow (2.34 seconds)	Yes	Yes	TMROF_EN=1	TMROF_STS
GPIO (Note 8)	Yes	Yes		
LAN_WAKE#	Yes	Yes	LAN_WAKE_EN=1	LAN_WAKE_STS
TCO SCI message from processor	Yes	No	None	TCOSCI_STS
TCO SCI Logic	Yes	No	TCOSCI_EN=1	TCOSCI_STS
TCO SMI Logic	No	Yes	TCO_EN=1	TCO_STS
TCO SMI –	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	SW_TCO_SMI
TCO SMI – Message from processor	No	Yes	None	OPISMI_STS
TCO SMI – NMI occurred (and NMIs mapped to SMI)	No	Yes	NMI2SMI_EN=1	NMI2SMI_STS
TCO SMI – INTRUDER# signal goes active	No	Yes	INTRD_SEL=10	INTRD_DET
TCO SMI – Change of the BIOSWE (D31:F0:DCh, Bit 0) bit from 0 to 1	No	Yes	BLE=1	BIOSWR_STS
TCO SMI – Write attempted to BIOS	No	Yes	BIOSWE=1	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	None	DEVTRAP_STS
SMBus Host Controller	No	Yes	SMB_SMI_EN Host Controller Enabled	SMBus host status reg.
SMBus Secondary SMI message	No	Yes	None	SMBUS_SMI_STS

Table 26-5. Causes of SMI and SCI (Sheet 2 of 2)

Cause	SCI	SMI	Additional Enables (Note 1)	Where Reported
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	SLP_SMI_EN=1	SLP_SMI_STS
SPI Command Completed	No	Yes	None	SPI_STS
eSPI SCI/SMI Request	Yes	Yes	eSPI_SCI_EN See eSPI section	eSPI_SCI_STS eSPI_SMI_STS
Software Generated GPE	Yes	Yes	SWGPE_EN=1	SWGPE_STS
Intel® ME	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 3.0 (xHCI) SMI Event	No	Yes	XHCI_SMI_EN=1	XHCI_SMI_STS
Wake Alarm Device Timer	Yes	Yes	WADT_EN	WADT_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

26.7.5.1 PCI Express* SCI

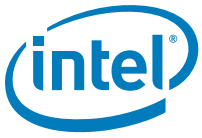
PCI Express ports and the processor have the ability to cause PME using messages. When a PME message is received, the PCH will set the PCI_EXP_STS bit. If the PCI_EXP_EN bit is also set, the PCH can cause an SCI using the GPE1_STS register.

26.7.5.2 PCI Express* Hot-Plug

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

26.7.6 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 processors impacts of C-states, such as voltage levels or processor clocking. In addition to the messages, the PCH also provides additional information to the processor using a sideband pin (PMSYNCH).



26.7.7 Dynamic 24-MHz Clock Control

The 24-MHz clock can be dynamically controlled independent of any other low-power state.

The Dynamic 24-MHz Clock control is handled using the following signal:

CLKRUN#: Used by LPC peripherals or other legacy devices to request the system 24-MHz clock to run.

26.7.7.1 Conditions for Checking the 24-MHz Clock

When there is a lack of activity, the PCH has the capability to stop the 24-MHz clocks to conserve power. "Clock activity" is defined as any activity that would require the 24-MHz clock to be running.

Any of the following conditions will indicate that it is **not okay** to stop the 24-MHz clock:

- Cycles on LPC
- SERIRQ activity

26.7.7.2 Conditions for Maintaining the 24-MHz Clock

LPC or any other devices that wish to maintain the 24-MHz clock running will observe the CLKRUN# signal de-asserted, and then must re-assert if (drive it low) within 92 clocks.

- When the PCH has tri-stated the CLKRUN# signal after de-asserting it, the PCH then checks to see if the signal has been re-asserted (externally).
- After observing the CLKRUN# signal asserted for 1 clock, the PCH again starts asserting the signal.
- If an internal device needs the PCI bus, the PCH asserts the CLKRUN# signal.

26.7.7.3 Conditions for Stopping the 24-MHz Clock

- When there is a lack of activity (as defined above) for ninety 24-MHz clock cycles, the PCH de-asserts (drive high) CLKRUN# for 1 clock and then tri-states the signal.
- If no device drives CLKRUN# low within 93 clock cycles after it has been de-asserted, the PCH will stop the 24-MHz clocks.

26.7.7.4 Conditions for Re-starting the 24-MHz Clock

- A peripheral asserts CLKRUN# to indicate that it needs the 24-MHz clock re-started.
- Observing the CLKRUN# signal asserted externally for 1 (free running) clock, the PCH again starts driving CLKRUN# asserted.

If an internal source requests the clock to be re-started, the PCH re-asserts CLKRUN#, then the PCH will start the 24-MHz clocks.

26.7.8 Sleep States

26.7.8.1 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.

26.7.8.2 Initiating Sleep State

Sleep states (S3–S5) are initiated by:

- Masking interrupts, turning off all primary bus 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.
- Shut down by integrated manageability functions (ASF/Intel AMT)
- Internal watchdog timer Timeout events

Table 26-6. 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#.

26.7.8.3 Exiting Sleep States

Sleep states (S3–S5) are exited based on wake events. The wake events force the system to a full on state (S0), although some non-critical subsystems might still be 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 26-7.

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.

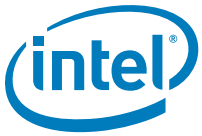


Table 26-7. Causes of Wake Events (Sheet 1 of 2)

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 register.	Yes	Yes	Yes	No
Power Button	Always enabled as Wake event.	Yes	Yes	Yes	Yes
Any GPIOs can be enabled for wake from the set of GPP_A to GPP_G and includes GPD[5]		Yes	No	No	No
LAN_WAKE#	Enabled natively (unless pin is configured to be in GPIO mode)	Yes	Yes	Yes	Yes
LAN	Will use PME#. Wake enables set with LAN logic.	Yes	No	Yes	No
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
PCI Express WAKE# pin	PCIEXPWAK_DIS bit.	Yes	Yes	Yes	No
SMBALERT#	(Note 4)	Yes	No	Yes	Yes
Secondary SMBus Wake Message (01h)	Wake/SMI# command always enabled as a Wake event. ***Note:*** Secondary SMBus 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 Secondary SMBus Command register. Reported in the SMB_WAK_STS bit in the GPE0_STS register.	Yes	No	Yes	Yes
Intel® ME 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	No	Yes	Yes
Wake Alarm Device	WADT_EN in GPE0_EN[127:96]	Yes	Yes	No	No

Table 26-7. 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)

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 ME-initiated power button override, Intel ME-initiated host partition reset with power down, Intel ME 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 statuses 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 primary status, "GPIO_TIER2_SCI_STS" or GPE0_STS[6Fh] and further comparison needed to know which 2-tier GPI(s) has triggered the GPIO Tier 2 SCI.

26.7.8.4 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.

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.

26.7.8.5 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.

The PCH monitors both PCH_PWROK and PCH_DPWROK to detect for power failures. If PCH_PWROK goes low, the PCHPWR_FLR bit is set. If PCH_DPWROK goes low, PWR_FLR is set.

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#.

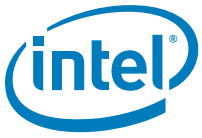


Table 26-8. Transitions Due to Power Failure

State at Power Failure	AFTERG3_EN Bit	Transition when Power Returns
S0, S3	1 0	S5 S0
S4	1 0	S4 S0
S5	1 0	S5 S0
Deep Sx	1 0	Deep Sx[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 S3, S4 or S5).
2. Power Failure is defined as PCH_PWROK or PCH_DPWROK transition low.

26.7.8.6 Deep Sx

To minimize power consumption while in S3/S4/S5, the PCH supports lower power, lower featured version of this 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.

26.7.8.6.1 Entry Into Deep Sx

A combination of conditions is required for entry into Deep Sx.

All of the following must be met:

1. Intel® ME in M-Off AND
2. Either a. or b. as defined below
   a. ((DPS3_EN_AC AND S3) OR (DPS4_EN_AC AND S4) OR (DPS5_EN_AC AND S5))
   b. ((ACPRESENT = 0) AND ((DPS3_EN_DC AND S3) OR (DPS4_EN_DC AND S4) OR (DPS5_EN_DC AND S5)))

Table 26-9. Supported Deep Sx Policy Configurations (Sheet 1 of 2)

Configuration	DPS3_EN_DC	DPS3_EN_AC	DPS4_EN_DC	DPS4_EN_AC	DPS5_EN_DC	DPS5_EN_AC
1. Enabled in S5 when on Battery (ACPRESENT = 0)	0	0	0	0	1	0
2. Enabled in S5 (ACPRESENT not considered)	0	0	0	0	1	1
3. Enabled in S4 and S5 when on Battery (ACPRESENT = 0)	0	0	1	0	1	0
4. Enabled in S4 and S5 (ACPRESENT not considered)	0	0	1	1	1	1
5. Enabled in S3, S4 and S5 when on Battery (ACPRESENT = 0)	1	0	1	0	1	0

Table 26-9. Supported Deep Sx Policy Configurations (Sheet 2 of 2)

Configuration	DPS3_EN_DC	DPS3_EN_AC	DPS4_EN_DC	DPS4_EN_AC	DPS5_EN_DC	DPS5_EN_AC
6. Enabled in S3, S4 and S5 (ACPRESENT not considered)	1	1	1	1	1	1
7. Deep S3/S4/ S5 disabled	0	0	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.

26.7.8.6.2 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 26-10. Deep Sx Wake Events

Event	Enable
RTC Alarm	RTC_DS_WAKE_DIS (RCBA+3318h:Bit 21)
Power Button	Always enabled
PCIe* WAKE# pin	PCIEXP_WAK_DIS
Wake Alarm Device	WADT_EN

ACPRESENT has some behaviors that are different from the other Deep Sx wake events. If the Intel® ME has enabled ACPRESENT as a wake event then it behaves just like any other Intel ME 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 S3/M-Off, S4/M-Off or S5/M-Off. If ACPRESENT subsequently drops (before any Host or Intel ME wake events are detected), the PCH will re-enter Deep Sx.

26.7.9 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.

26.7.9.1 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 26-11.

After any PWRBTN# assertion (falling edge), subsequent falling PWRBTN# edges are ignored until after 16ms if PM_CFG.PB_DB_MODE='0' or after 500us if PM_CFG.PB_DB_MODE='1'.

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.

Table 26-11. 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 consecutive seconds	Unconditional transition to S5 state and if Deep Sx is enabled and conditions are met per Section 26.7.8.6, the system will then transition to Deep Sx.	No dependence on processor or any other subsystem

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.

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 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 debounce. If PM_CFG.PB_DB_MODE='1', the status is taken before the debounce. 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. See the *Advanced Configuration and Power Interface Specification* for implementation details.

26.7.9.2 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 bit. This is separate from the external PME# signal and can cause the same effect.

26.7.9.3 SYS_RESET# Signal

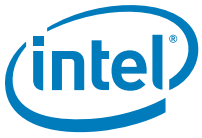
When the SYS_RESET# pin is detected as active after the 16 ms debounce logic, the PCH attempts to perform a "graceful" reset by entering a host partition reset entry sequence.

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 debounce logic, and the system is back to a full S0 state with PLTRST# inactive.

Note: If bit 3 of the CF9h I/O register is set 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.



26.7.9.4 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 looks 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.

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 state machine logic to perform the power down, the state machine may not be working, and the system will not power down.

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.

26.7.9.5 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 DeepSx->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/DeepSx or in the process of exiting G3/DeepSx (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 DeepSx 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.

26.7.10 ALT Access Mode

Before entering a low-power state, several registers from powered down parts may need to be saved. In the majority of cases, this is not an issue, as registers have read and write paths. However, several of the ISA compatible registers are either read only or write only. To get data out of write-only registers, and to restore data into read-only registers, the PCH implements an ALT access mode.

If the ALT access mode is entered and exited after reading the registers of the PCH timer (8254), the timer starts counting faster (13.5 ms). The following steps listed below can cause problems:

1. BIOS enters ALT access mode for reading the PCH timer-related registers.
2. BIOS exits ALT access mode.
3. BIOS continues through the execution of other needed steps and passes control to the operating system.

After getting control in step #3, if the operating system does not reprogram the system timer again, the timer ticks may be happening faster than expected.

Operating systems reprogram the system timer and therefore do not encounter this problem.

For other operating systems, the BIOS should restore the timer back to 54.6 ms before passing control to the operating system. If the BIOS is entering ALT access mode before entering the suspend state it is not necessary to restore the timer contents after the exit from ALT access mode.

26.7.10.1 Write-Only Registers with Read Paths in ALT Access Mode

The registers described in Table 26-12 have read paths in ALT access mode. The access number field in the table indicates which register will be returned per access to that port.

Table 26-12. Write-Only Registers with Read Paths in ALT Access Mode

Restore Data				Restore Data			
I/O Addr	# of Rds	Access	Data	I/O Addr	# of Rds	Access	Data
20h	12	1	PIC ICW2 of Primary controller	40h	7	1	Timer Counter 0 status, bits [5:0]
		2	PIC ICW3 of Primary controller			2	Timer Counter 0 base count low byte
		3	PIC ICW4 of Primary controller			3	Timer Counter 0 base count high byte
		4	PIC OCW1 of Primary controller[1]			6	Timer Counter 2 base count low byte
		5	PIC OCW2 of Primary controller			7	Timer Counter 2 base count high byte
		6	PIC OCW3 of Primary controller	42h	1		Timer Counter 2 status, bits [5:0]
		7	PIC ICW2 of Secondary controller	70h	1		Bit 7 = NMI Enable, Bits [6:0] = RTC Address
		8	PIC ICW3 of Secondary controller	70h	1		Bit 7 = Read value is '0'. Bits [6:0] = RTC Address
		9	PIC ICW4 of Secondary controller				
		10	PIC OCW1 of Secondary controller[1]				
		11	PIC OCW2 of Secondary controller				
		12	PIC OCW3 of Secondary controller				

Notes:
1. The OCW1 register must be read before entering ALT access mode.
2. Bits 5, 3, 1, and 0 return 0.

26.7.10.2 PIC Reserved Bits

Many bits within the PIC are reserved, and must have certain values written in order for the PIC to operate properly. Therefore, there is no need to return these values in ALT access mode. When reading PIC registers from 20h and A0h, the reserved bits shall return the values listed in Table 26-13.

Table 26-13. PIC Reserved Bits Return Values (Sheet 1 of 2)

PIC Reserved Bits	Value Returned
ICW2(2:0)	000
ICW4(7:5)	000
ICW4(3:2)	00
ICW4(0)	0

Table 26-13. PIC Reserved Bits Return Values (Sheet 2 of 2)

PIC Reserved Bits	Value Returned
OCW2(4:3)	00
OCW3(7)	0
OCW3(5)	Reflects bit 6
OCW3(4:3)	01

26.7.10.3 Read Only Registers with Write Paths in ALT Access Mode

The registers described in Table 26-14 have write paths to them in ALT access mode. Software restores these values after returning from a powered down state. These registers must be handled special by software. When in normal mode, writing to the base address/count register also writes to the current address/count register. Therefore, the base address/count must be written first, then the part is put into ALT access mode and the current address/count register is written.

Table 26-14. Register Write Accesses in ALT Access Mode

I/O Address	Register Write Value
08h	DMA Status Register for Channels 0–3
D0h	DMA Status Register for Channels 4–7

26.7.11 System Power Supplies, Planes, and Signals

26.7.11.1 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 power 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.

The SLP_S4# output signal is used to remove power to additional subsystems that are powered during SLP_S3#.

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_A# output signal can be used to cut power to the Intel Management Engine and SPI flash on a platform that supports the M3 state (for example, certain power policies in Intel AMT).

SLP_LAN# output signal can be used to cut power to the external Intel 82579 GbE PHY device.

26.7.11.2 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.

26.7.11.3 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.

26.7.11.4 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.

26.7.11.5 SLP_LAN# Pin Behavior

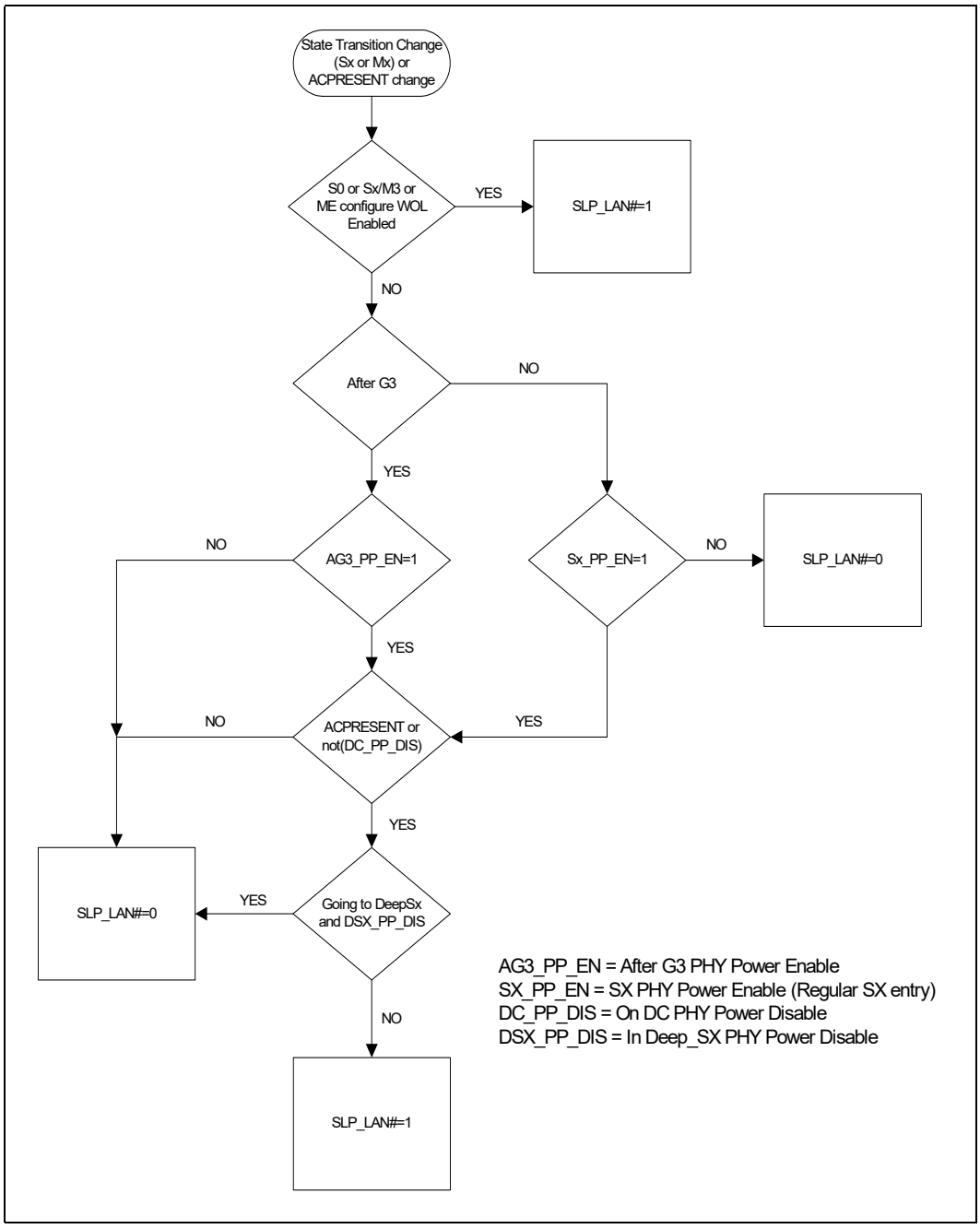
The PCH controls the voltage rails into the external LAN PHY using the SLP_LAN# pin.

- The LAN PHY is always powered when the Host and Intel® ME systems are running.
  - SLP_LAN#='1' whenever SLP_S3#='1' or SLP_A#='1'.
- If the LAN PHY is required by Intel ME in Sx/M-Off or Deep Sx, Intel ME must configure SLP_LAN#='1' irrespective of the power source and the destination power state. Intel ME 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 (B0:D31:F0:A0h bit 28).
- If the LAN PHY is required in Sx/M-Off, the host BIOS must set SX_PP_EN (B0:D31:F0:A0h bit 27).
- If the LAN PHY is required in Deep Sx, the host BIOS must keep DSX_PP_DIS (B0:D31:F0:A0h bit 29) cleared.
- If the LAN PHY is not required if the source of power is battery, the host BIOS must set DC_PP_DIS (B0:D31:F0:A0h bit 30).

Note: Intel® ME configuration of SLP_LAN# in Sx/M-Off and Deep Sx is dependent on Intel ME 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 26-1. Conceptual Diagram of SLP_LAN#



26.7.11.6 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 (RCBA+3318h bit 4).
- If Intel ME has access to the Wireless LAN device:
  - The Wireless LAN device must always be powered as long as Intel ME is powered. SLP_WLAN#='1' whenever SLP_A#='1'.
  - If Wake on Wireless LAN (WoWLAN) is required from M-Off state, Intel ME will configure SLP_WLAN#='1' in Sx/M-Off.

Intel® ME configuration of SLP_WLAN# in Sx/M-Off is dependent on Intel ME power policy configuration.

26.7.11.7 SUSPWRDNACK/SUSWARN#/GPP_A13 Steady State Pin Behavior

Table 26-15 summarizes SUSPWRDNACK/SUSWARN#/GPP_A13 pin behavior.

Table 26-15. SUSPWRDNACK/SUSWARN#/GPP_A13 Pin Behavior

Pin	Deep Sx (Supported /Not-Supported)	GPP_A13 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® ME power package and power source (Note 1)	0	Off
SUSWARN#	Supported	Native	1	1 (Note 2)	1	Off
GPP_A13	Don't Care	IN	High-Z	High-Z	High-Z	Off
	Don't Care	OUT	Depends on GPP_A13 output data value	Depends on GPP_A13 output data value	Depends on GPP_A13 output data value	Off

Notes:
1. PCH will drive SPDA pin based on Intel ME power policy configuration.
2. If entering Deep Sx, pin will assert and become undriven ("Off") when suspend well drops upon Deep Sx entry.

Table 26-16. 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 ME power policy configuration.

Note: See Table 26-17

26.7.11.8 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.

26.7.12 Legacy Power Management Theory of Operation

Instead of relying on ACPI software, legacy power management uses BIOS and various hardware mechanisms. The scheme relies on the concept of detecting when individual subsystems are idle, detecting when the whole system is idle, and detecting when accesses are attempted to idle subsystems.

However, the operating system is assumed to be at least APM enabled. Without APM calls, there is no quick way to know when the system is idle between keystrokes. The PCH does not support burst modes.

26.7.12.1 Mobile APM Power Management

In mobile systems, there are additional requirements associated with device power management. To handle this, the PCH has specific SMI traps available. The following algorithm is used:

1. The periodic SMI timer checks if a device is idle for the require time. If so, it puts the device into a low-power state and sets the associated SMI trap.
2. When software (not the SMI handler) attempts to access the device, a trap occurs (the cycle does not really go to the device and an SMI is generated).
3. The SMI handler turns on the device and turns off the trap.
4. The SMI handler exits with an I/O restart. This allows the original software to continue.

26.7.13 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 as defined in Chapter 9.

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 see Table 26-17 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® ME 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® ME 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 26-17 shows the various reset triggers.

Table 26-17. 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)	
Secondary SMBus Message received for Reset with Power-Cycle	No	Yes	No (Note 4)	
Secondary SMBus Message received for Reset without Power-Cycle	Yes	No	No (Note 4)	
Secondary SMBus 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

Table 26-17. 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)
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	Yes
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	Yes
Intel® Management Engine Hardware Uncorrectable Error	No	No	No	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 26.7.8.6, 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.

§ §

27 Real Time Clock (RTC)

27.1 Acronyms

Acronyms	Description
GPI	General Purpose Input
RAM	Random Access Memory
RTC	Real Time Clock

27.2 References

None

27.3 Overview

The PCH contains a Motorola MC146818B-compatible real-time clock with 256 bytes of battery-backed RAM. The real-time clock performs two key functions—keeping track of the time of day and storing system data, even when the system is powered down. The RTC operates on a 32.768-KHz crystal and a 3V battery.

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 30 days in advance, rather than just 24 hours in advance.

27.4 Signal Description

Name	Type	Description
RTCX1	I	**Crystal Input 1:** This signal is connected to the 32.768-KHz crystal. If no external crystal is used, then RTCX1 can be driven with the desired clock rate. Maximum voltage allowed on this pin is 1.2V.
RTCX2	O	**Crystal Input 2:** This signal is connected to the 32.768-KHz crystal. 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), 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.
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.

27.5 Integrated Pull-Ups and Pull-Downs

None

27.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
RTCRST#	RTC	Undriven	Undriven	Undriven	Undriven
SRTCRST#	RTC	Undriven	Undriven	Undriven	Undriven

27.7 Functional Description

The Real Time Clock (RTC) module provides a battery backed-up date and time keeping device with two banks of static RAM with 128 bytes each, although the first bank has 114 bytes for general-purpose usage.

Three interrupt features are available: time of day alarm with once a second to once a month range, periodic rates of 122 – 500 ms, and end of update cycle notification. Seconds, minutes, hours, days, day of week, month, and year are counted. Daylight savings compensation is no longer supported.

The hour is represented in twelve or twenty-four hour format, and data can be represented in BCD or binary format. The design is functionally compatible with the Motorola MS146818B. The time keeping comes from a 32.768-KHz oscillating source, which is divided to achieve an update every second. The lower 14 bytes on the lower RAM block has very specific functions. The first ten are for time and date information. The next four (0Ah to 0Dh) are registers, which configure and report RTC functions.

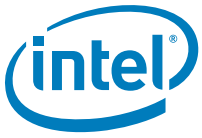
The time and calendar data should match the data mode (BCD or binary) and hour mode (12 or 24 hour) as selected in register B. It is up to the programmer to make sure that data stored in these locations is within the reasonable values ranges and represents a possible date and time. The exception to these ranges is to store a value of C0–FFh in the Alarm bytes to indicate a don't care situation. All Alarm conditions must match to trigger an Alarm Flag, which could trigger an Alarm Interrupt if enabled.

The SET bit must be 1 while programming these locations to avoid clashes with an update cycle. Access to time and date information is done through the RAM locations. If a RAM read from the ten time and date bytes is attempted during an update cycle, the value read does not necessarily represent the true contents of those locations. Any RAM writes under the same conditions are ignored.

Note: The leap year determination for adding a 29th day to February does not take into account the end-of-the-century exceptions. The logic simply assumes that all years divisible by 4 are leap years. According to the Royal Observatory Greenwich, years that are divisible by 100 are typically not leap years. In every fourth century (years divisible by 400, like 2000), the 100-year-exception is over-ridden and a leap-year occurs.

Note: The year 2100 will be the first time in which the current RTC implementation would incorrectly calculate the leap-year.

The PCH does not implement month/year alarms.



27.7.1 Update Cycles

An update cycle occurs once a second, if the SET bit of register B is not asserted and the divide chain is properly configured. During this procedure, the stored time and date are incremented, overflow is checked, a matching alarm condition is checked, and the time and date are rewritten to the RAM locations.

The update cycle will start at least 488 µs after the UIP bit of register A is asserted, and the entire cycle does not take more than 1984 µs to complete. The time and date RAM locations (0–9) are disconnected from the external bus during this time.

To avoid update and data corruption conditions, external RAM access to these locations can safely occur at two times. When an updated-ended interrupt is detected, almost 999 ms is available to read and write the valid time and date data. If the UIP bit of Register A is detected to be low, there is at least 488 µs before the update cycle begins.

Warning: The overflow conditions for leap years adjustments are based on more than one date or time item. To ensure proper operation when adjusting the time, the new time and data values should be set at least two seconds before leap year occurs.

27.7.2 Interrupts

The real-time clock interrupt is internally routed within the PCH both to the I/O APIC and the 8259. It is mapped to interrupt vector 8. This interrupt does not leave the PCH, nor is it shared with any other interrupt. IRQ8# from the SERIRQ stream is ignored. However, the High Performance Event Timers can also be mapped to IRQ8#; in this case, the RTC interrupt is blocked.

27.7.3 Lockable RAM Ranges

The RTC battery-backed RAM supports two 8-byte ranges that can be locked using the configuration space. If the locking bits are set, the corresponding range in the RAM will not be readable or writable. A write cycle to those locations will have no effect. A read cycle to those locations will not return the location’s actual value (resultant value is undefined).

Once a range is locked, the range can be unlocked only by a hard reset, which will invoke the BIOS and allow it to relock the RAM range.

27.7.4 Century Rollover

The PCH detects a rollover when the Year byte transitions form 99 to 00. Upon detecting the rollover, the PCH sets the NEWCENTURY_STS bit.

If the system is in an S0 state, this causes an SMI#. The SMI# handler can update registers in the RTC RAM that are associated with century value.

If the system is in a sleep state (S3–S5) when the century rollover occurs, the PCH also sets the NEWCENTURY_STS bit, but no SMI# is generated. When the system resumes from the sleep state, BIOS should check the NEWCENTURY_STS bit and update the century value in the RTC RAM.

27.7.5 Clearing Battery-Backed RTC RAM

Clearing CMOS RAM in a PCH-based platform can be done by using a jumper on RTCRST# or GPI. Implementations should not attempt to clear CMOS by using a jumper to pull VccRTC low.

27.7.5.1 Using RTCRST# to Clear CMOS

A jumper on RTCRST# can be used to clear CMOS values, as well as reset to default, the state of those configuration bits that reside in the RTC power well.

When the RTCRST# is strapped to ground, the RTC_PWR_STS bit will be set and those configuration bits in the RTC power well will be set to their default state. BIOS can monitor the state of this bit and manually clear the RTC CMOS array once the system is booted. The normal position would cause RTCRST# to be pulled up through a weak Pull-up resistor. This RTCRST# jumper technique allows the jumper to be moved and then replaced—all while the system is powered off. Then, once booted, the RTC_PWR_STS can be detected in the set state.

27.7.5.2 Using a GPI to Clear CMOS

A jumper on a GPI can also be used to clear CMOS values. BIOS would detect the setting of this GPI on system boot-up, and manually clear the CMOS array.

Note: The GPI strap technique to clear CMOS requires multiple steps to implement. The system is booted with the jumper in new position, then powered back down. The jumper is replaced back to the normal position, then the system is rebooted again.

Warning: Do not implement a jumper on VccRTC to clear CMOS.

27.7.6 External RTC Circuitry

The PCH implements an internal oscillator circuit that is sensitive to step voltage changes in VCCRTC.

Table 27-1. RTC Crystal Requirements

Parameter	Specification
Frequency	32.768 KHz
Typical Tolerance	20 ppm or better
ESR	≤ 50 KΩ

Table 27-2. External Crystal Oscillator Requirements

Parameter	Specification
Frequency	32.768 KHz
Typical Tolerance	20 ppm or better
Voltage Swing	0 to 1.0Vp-p (±5%)

§ §

28 Serial ATA (SATA)

The PCH has an integrated Serial ATA (SATA) host controller with independent DMA operation on up to two ports for PCH-Y; up to three ports for PCH-U.

28.1 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

28.2 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

28.3 Overview

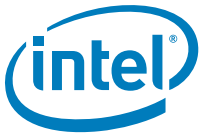
The PCH has one integrated SATA host controller that supports independent DMA operation for up to two ports for PCH-Y and up to three ports for PCH-U and supports data transfer rates of up to 6 Gb/s on all ports.

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.

28.4 Signal Description

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.
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_E6	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/ PCIE7_TXP **SATA0_TXN**/ PCIE7_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 7 signals. ***Note:*** Use FITC to set the soft straps of the SATA/PCIe* Combo Port 0 Strap (PCIE_SATA_P0_Flex) that select this port as SATA Port 0 or PCIe* Port 7. The default SATA/PCIe* port assignment is PCIe* Port 7. ***Note:*** When PCIE_SATA_P0_Flex=11, the assignment of the SATA Port 0 versus PCIe* Port 7 will be based on the polarity of SATAXPCIE0. Use FITC to set the soft strap of the Polarity Select SATA/PCIe* Combo Port 0 (PSCPSP_P0_STRP).
SATA0_RXP/ PCIE7_RXP **SATA0_RXN**/ PCIE7_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 7 signals. ***Note:*** Use FITC to set the soft straps of the SATA/PCIe* Combo Port 0 Strap (PCIE_SATA_P0_Flex) that select this port as SATA Port 0 or PCIe* Port 7. The default SATA/PCIe* port assignment is PCIe* Port 7. ***Note:*** When PCIE_SATA_P0_Flex=11, the assignment of the SATA Port 0 versus PCIe* Port 7 will be based on the polarity of SATAXPCIE0. Use FITC to set the soft strap of the Polarity Select SATA/PCIe* Combo Port 0 (PSCPSP_P0_STRP).
SATA1A_TXP/ PCIE8_TXP **SATA1A_TXN**/ PCIE8_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 8 signals. ***Note:*** **For PCH-Y**, the SATA Port 1 can be configured to PCIe* Port 8. **For PCH-U Only,** the SATA Port 1 can be configured to PCIe* Port 8 or Port 11. ***Note:*** Use FITC to set the soft straps of the SATA/PCIe* Combo Port 1 Strap (PCIE_SATA_P1_Flex) that select this port as SATA Port 1 or PCIe* Port 8. The default SATA/PCIe* port assignment is PCIe* Port 8. ***Note:*** When PCIE_SATA_P1_Flex=11, the assignment of the SATA Port 1 versus PCIe* Port 8 will be based on the polarity of SATAXPCIE1. Use FITC to set the soft strap of the Polarity Select SATA/PCIe* Combo Port 1 (PSCPSP_P1_STRP).



Name	Type	Description
SATA1A_RXP/ PCIE8_RXP **SATA1A_RXN**/ PCIE8_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 8 signals. ***Note:*** **For PCH-Y**, the SATA Port 1 can be configured to PCIe* Port 8. **For PCH-U Only**, the SATA Port 1 can be configured to PCIe* Port 8 or Port 11. ***Note:*** Use FITC to set the soft straps of the SATA/PCIe* Combo Port 1 Strap (PCIE_SATA_P1_Flex) that select this port as SATA Port 1 or PCIe* Port 8. The default SATA/PCIe* port assignment is PCIe* Port 8. ***Note:*** When PCIE_SATA_P1_Flex=11, the assignment of the SATA Port 1 versus PCIe* Port 8 will be based on the polarity of SATAXPCIE1. Use FITC to set the soft strap of the Polarity Select SATA/PCIe* Combo Port 1 (PSCPSP_P1_STRP).
SATA1B_TXP/ PCIE11_TXP **SATA1B_TXN**/ PCIE11_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 11 signals. ***Note:*** **For PCH-U Only,** the SATA Port 1 can be configured to PCIe* Port 8 or Port 11. ***Note:*** Use FITC to set the soft straps of the SATA/PCIe* Combo Port 2 Strap (PCIE_SATA_P2_Flex) that select this port as SATA Port 1 or PCIe* Port 11. The default SATA/PCIe* port assignment is PCIe* Port 11. ***Note:*** When PCIE_SATA_P2_Flex=11, the assignment of the SATA Port 1 versus PCIe* Port 11 will be based on the polarity of SATAXPCIE1. Use FITC to set the soft strap of the Polarity Select SATA/PCIe* Combo Port 2 (PSCPSP_P2_STRP).
SATA1B_RXP/ PCIE11_RXP **SATA1B_RXN**/ PCIE11_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 11 signals. ***Note:*** **For PCH-U Only**, the SATA Port 1 can be configured to PCIe* Port 8 or Port 11. ***Note:*** Use FITC to set the soft straps of the SATA/PCIe* Combo Port 2 Strap (PCIE_SATA_P2_Flex) that select this port as SATA Port 1 or PCIe* Port 11. The default SATA/PCIe* port assignment is PCIe* Port 11. ***Note:*** When PCIE_SATA_P2_Flex=11, the assignment of the SATA Port 1 versus PCIe* Port 11 will be based on the polarity of SATAXPCIE1. Use FITC to set the soft strap of the Polarity Select SATA/PCIe* Combo Port 2 (PSCPSP_P2_STRP).
SATA2_TXP/ PCIE12_TXP **SATA2_TXN**/ PCIE12_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 12 signals. ***Note:*** **For PCH-U only:** Use FITC to set the soft straps of the SATA/PCIe* Combo Port 3 Strap (PCIE_SATA_P3_Flex) that select this port as SATA Port 2 or PCIe* Port 12. The default SATA/PCIe* port assignment is PCIe* Port 12. ***Note:*** When PCIE_SATA_P3_Flex=11, the assignment of the SATA Port 2 versus PCIe* Port 12 will be based on the polarity of SATAXPCIE2. Use FITC to set the soft strap of the Polarity Select SATA/PCIe* Combo Port 3 (PSCPSP_P3_STRP).
SATA2_RXP/ PCIE12_RXP **SATA2_RXN**/ PCIE12_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 12 signals. ***Note:*** **For PCH-U only:** Use FITC to set the soft straps of the SATA/PCIe* Combo Port 3 Strap (PCIE_SATA_P3_Flex) that select this port as SATA Port 2 or PCIe* Port 12. The default SATA/PCIe* port assignment is PCIe* Port 12. ***Note:*** When PCIE_SATA_P3_Flex=11, the assignment of the SATA Port 2 versus PCIe* Port 12 will be based on the polarity of SATAXPCIE2. Use FITC to set the soft strap of the Polarity Select SATA/PCIe* Combo Port 3 (PSCPSP_P3_STRP).

Name	Type	Description
SATAGP0/ SATAXPCIE0/ GPP_E0	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/ SATAXPCIE1/ GPP_E1	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.
SATAGP2/ SATAXPCIE2/ GPP_E2	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	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.

28.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Nominal Value	Notes
SATAXPCIE[2:0]	Pull-up	20 KΩ	1, 2

Note:
1. SATAGP[2:0]/SATAXPCIE[2:0]/GPP_E[2:0] has two native functions – the first native function (SATAXPCIEx) is selected if the Flex I/O soft strap PCIE_SATA_Px_Flex = 11b. Setting PCIE_SATA_Px_Flex = 11b also enables an internal Pull-up resistor in this pin to allow Flexible I/O selection of SATA Port x or PCIe* Port x to be assigned based on the type of card installed and based on the SATAXPCIEx mux selector with the polarity for SATA or PCIe* (When PSCPSP_Px_STRP = 0, PCIe* will be selected if the sampled value is "0" and SATA will be selected if the sampled value is "1"; When PSCPSP_Px_STRP = 1, SATA will be selected if the sampled value is "0" and PCIe* will be selected if the sampled value is "1"). Use FITC to set the soft straps of the PCIe/SATA Combo Port x Strap (PCIE_SATA_Px_Flex) and Polarity Select SATA/PCIe* Combo Port x (PSCPSP_Px_STRP). .
2. Simulation data shows that these resistor values can range from 14 KΩ – 26 KΩ.

28.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	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_E8[1]	Primary	Undriven	Undriven	Undriven	Off
DEVSLP[2:0]/ GPP_E[6:4][1]	Primary	Undriven	Undriven	Undriven	Off

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
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.

28.7 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 features two ports for PCH-Y and three ports for PCH-U that can be independently enabled or disabled (they cannot be tri-stated or driven low). Each interface is supported by an independent DMA controller.

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.

28.7.1 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.

28.7.2 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)

28.7.3 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.

28.7.4 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.

28.7.4.1 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.

28.7.4.2 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.

28.7.5 Intel® Smart Response Technology

Intel® Smart Response Technology is a disk caching solution that can provide improved computer system performance with improved power savings. It allows configuration of a computer system with the advantage of having HDDs for maximum storage capacity with system performance at or near SSD performance levels.

Part of the Intel® RST storage class driver feature set, Intel® Smart Response Technology implements storage I/O caching to provide users with faster response times for things like system boot and application startup. On a traditional system, performance of these operations is limited by the hard drive, particularly when there may be other I/O intensive background activities running simultaneously, like system updates or virus scans. Intel® Smart Response Technology accelerates the system response experience by putting frequently-used blocks of disk data on an SSD, providing dramatically faster access to user data than the hard disk alone can provide. The user sees the full capacity of the hard drive with the traditional single drive letter with overall system responsiveness similar to what an SSD-only system provides.

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.

28.7.6 Power Management Operation

Power management of the PCH SATA controller and ports will cover operations of the host controller and the SATA link.

28.7.6.1 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 is 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 (see 13.7.9.1, 13.7.9.4 of the 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.

28.7.6.2 Power State Transitions

28.7.6.2.1 Partial and Slumber State Entry/Exit

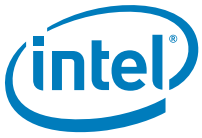
The partial and slumber states save interface power when the interface is idle. It would be most analogous to CLKRUN# (in power savings, not in mechanism), where the interface can have power saved while no commands are pending. The SATA controller defines PHY layer power management (as performed using 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.

28.7.6.2.2 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.

28.7.6.2.3 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 than sending commands over the interface to the device. The command most likely to be used in ATA/ATAPI is the "STANDBY IMMEDIATE" command.

28.7.6.2.4 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 primary 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.

28.7.6.3 Low-Power Platform Consideration

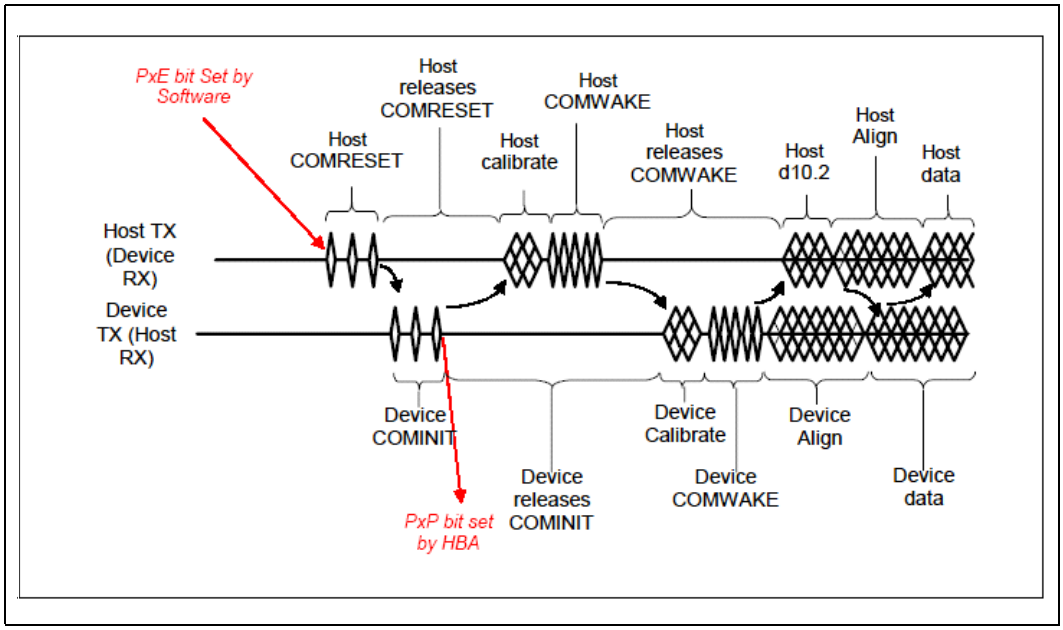
When low-power feature is enabled, the Intel SATA controller may power off PLLs or OOB detection circuitry while in the Slumber link power state. As a result, a device initiated wake may not be recognized by the host. For example, when the low-power feature is enabled it can prevent a Zero Power ODD (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.

28.7.7 SATA Device Presence

The flow used to indicate SATA device presence is shown in Figure 28-1. 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 see 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 see if a new device is connected.

Figure 28-1. Flow for Port Enable/Device Present Bits



28.7.8 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.

28.7.9 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 primary—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 primary/secondary designation for SATA devices—each device is treated as a primary—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. See Section 7.3.1 of the *AHCI Specification* for more information.

28.7.10 External SATA

The PCH SATA controller supports external SATA. External SATA utilizes the SATA interface outside of the system box. The usage model for this feature must comply with the Serial ATA II (SATA 3Gb/s) Cables and Connectors Volume 2 Gold specification at: www.sata-io.org. Intel validates one configuration:

- The back-panel solution involves running a trace to the I/O back panel and connecting a device using an external SATA connector on the board.

§ §

29 System Management Interface and SMLink

29.1 Acronyms

Acronyms	Description
BMC	Baseboard Management Controller
EC	Embedded Controller
NFC	Near Field Communication

29.2 References

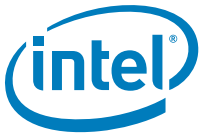
None

29.3 Overview

The PCH provides two SMLink interfaces, SMLink0 and SMLink1. The interfaces are intended for system management and are controlled by the Intel® ME. See the System Management chapter for more detail.

29.4 Signal Description

Name	Type	Description
INTRUDER#	I	**Intruder Detect:** This signal can be set to disable the system if box detected open.
SML0DATA/ GPP_C4	I/OD	**System Management Link 0 Data:** SMBus link to external PHY. External Pull-up is required.
SML0BDATA/ GPP_D13/ ISH_UART0_RXD/ I2C4B_SDA	I/OD	**Second Instant of System Management Link 0 Data: used for Comms Hub.** External Pull-up is required.
SML0BCLK/ GPP_D14/ ISH_UART0_TXD/ I2C4B_SCL	I/OD	**Second Instant of System Management Link 0 Clock: used for Comms Hub.** External Pull-up is required.
SML1CLK/GPP_C6	I/OD	**System Management Link 1 Clock:** SMBus link to optional Embedded Controller or BMC. External Pull-up resistor is required.
SML1DATA/ GPP_C7	I/OD	**System Management Link 1 Data:** SMBus link to optional Embedded Controller or BMC. External Pull-up resistor is required.
SML1ALERT#/ PCHHOT#/GPP_B23	I/OD	**System Management 1 Alert:** Alert for the Intel ME SMBus controller to optional Embedded Controller or BMC. A soft-strap determines the native function SML1ALERT# or PCHHOT# usage. External Pull-up resistor is required on this pin.



29.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
SML[1:0]ALERT#	Pull-down	14K - 26K	The internal Pull-down resistor is disable after RSMRST# de-asserted

29.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
INTRUDER#	RTC	Undriven	Undriven	Undriven	Off
SML[1:0]DATA	Primary	Undriven	Undriven	Undriven	Off
SML[1:0]CLK	Primary	Undriven	Undriven	Undriven	Off
SML[1:0]ALERT#	Primary	Internal Pull-down	Driven Low	Internal Pull-down	Off

29.7 Functional Description

The SMLink interfaces are controlled by the Intel® ME.

SMLink0 is mainly used for integrated LAN and NFC. 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.

§ §

30 Host System Management Bus (SMBus) Controller

30.1 Acronyms

Acronyms	Description
ARP	Address Resolution Protocol
CRC	Cyclic Redundancy Check
PEC	Package Error Checking
SMBus	System Management Bus

30.2 References

Specification	Location
System Management Bus (SMBus) Specification, Version 2.0	http://www.smbus.org/specs/

30.3 Overview

The PCH provides a System Management Bus (SMBus) 2.0 host controller as well as a Secondary SMBus Interface. The PCH is also capable of operating in a mode in which it can communicate with I^2C compatible devices.

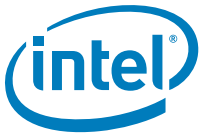
The host SMBus controller supports up to 100-KHz clock speed.

30.4 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.

30.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
SMBALERT#	Pull-down	9K - 50K	The integrated pull-down is disabled after RSMRST# de-assertion.



30.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
SMBDATA	Primary	Undriven	Undriven	Undriven	Off
SMBCLK	Primary	Undriven	Undriven	Undriven	Off
SMBALERT#	Primary	Internal Pull-down	Driven Low	Driven Low	Off

30.7 Functional Description

The PCH provides a System Management Bus (SMBus) 2.0 host controller as well as an Secondary SMBus Interface.

- **Host Controller:** Provides a mechanism for the processor to initiate communications with SMBus peripherals (secondaries). The PCH is also capable of operating in a mode in which it can communicate with I^2C compatible devices.
- **Secondary Interface:** Allows an external primary 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 Secondary Interface.

30.7.1 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 programming model of the host controller is combined into two portions: a PCI configuration portion, and a system I/O mapped portion. All static configurations, such as the I/O base address, is done using the PCI configuration space. Real-time programming of the Host interface is done in system I/O space.

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. If bit 6 and bit 8 of the PCI Command Register are set, an SERR# is generated and the signaled SERR# bit in the PCI Status Register is set.

30.7.1.1 Host Controller Operation Overview

The SMBus host controller is used to send commands to other SMBus secondary 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 (see *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, Block Write–Block Read Process Call, and Host Notify.

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 Secondary 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.

Secondary functionality, including the Host Notify protocol, is available on the SMBus pins.

Using the SMB host controller to send commands to the PCH secondary SMB port is not supported.

30.7.1.2 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.

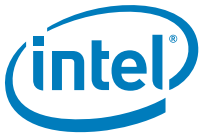
Quick Command

When programmed for a Quick Command, the Transmit Secondary 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. See 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 Secondary Address and Device Command Registers are sent. For the Receive Byte command, the Transmit Slave Secondary 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. See sections 5.5.2 and 5.5.3 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.



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 Secondary 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. See 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 secondary device. Then it must follow that command with a repeated start condition to denote a read from that device's address. The secondary 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 Secondary 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. See 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 secondary 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 Secondary 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. See 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 Secondary 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 secondary 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 secondary 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 secondary 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 Secondary 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. See 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. Also, if the Block Write protocol sequence changes slightly, the Byte Count (bits 27:20 in the bit sequence) are not sent. As a result, the secondary will not acknowledge (bit 28 in the sequence).

Note: 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 Secondary Address Register (SMB I/O register, offset 04h) needs to be 0.

The format that is used for the command is shown in Table 30-1.

Table 30-1. I^2C* Block Read (Sheet 1 of 2)

Bit	Description
1	Start
8:2	Secondary Address – 7 bits
9	Write
10	Acknowledge from secondary
18:11	Send DATA1 register
19	Acknowledge from secondary

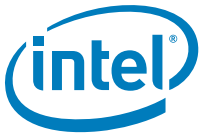


Table 30-1. I²C* Block Read (Sheet 2 of 2)

Bit	Description
20	Repeated Start
27:21	Secondary Address – 7 bits
28	Read
29	Acknowledge from secondary
37:30	Data byte 1 from secondary – 8 bits
38	Acknowledge
46:39	Data byte 2 from secondary – 8 bits
47	Acknowledge
–	Data bytes from secondary/Acknowledge
–	Data byte N from secondary – 8 bits
–	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 secondary 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 primary 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 secondary 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 secondary 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.

See section 5.5.8 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

30.7.1.3 Bus Arbitration

Several primaries 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 primary 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.

When the PCH is a primary SMBus, it drives the clock. When the PCH is sending address or command as an primary SMBus, or data bytes as a primary on writes, it drives data relative to the clock it is also driving. It will not start toggling the clock until the start or stop condition meets proper setup and hold time. The PCH will also ensure minimum time between SMBus transactions as a primary.

Note: The PCH supports the same arbitration protocol for both the SMBus and the System Management (SMLink) interfaces.

30.7.1.4 Clock Stretching

Some devices may not be able to handle their clock toggling at the rate that the PCH as an primary SMBus 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 primary SMBus if it is not ready to send or receive data.

30.7.1.5 Bus Timeout (PCH as Primary SMBus)

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 last 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).

30.7.1.6 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.

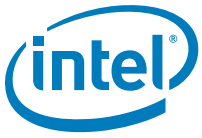


Table 30-2, Table 30-3 and Table 30-4 specify how the various enable bits in the SMBus function control the generation of the interrupt, Host and Secondary SMI, and Wake internal signals. The rows in the tables are additive, which means that if more than one row is true for a particular scenario then the Results for all of the activated rows will occur.

Table 30-2. 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 (Secondary 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	Secondary SMI# generated (SMBUS_SMI_STS)
	1	0	0	Interrupt generated

Table 30-3. Enables for SMBus Secondary 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
Secondary Write to Wake/ SMI# Command	X	X	Wake generated when asleep. Secondary SMI# generated when awake (SMBUS_SMI_STS).
Secondary Write to SMLINK_SLAVE_SMI Command	X	X	Secondary 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 30-4. Enables for the Host Notify Command

HOST_NOTIFY_INTREN (Secondary Control I/O Register, Offset 11h, Bit 0)	SMB_SMI_EN (Host Config Register, D31:F4:Off40h, Bit 1)	HOST_NOTIFY_WKEN (Secondary 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	Secondary SMI# generated (SMBUS_SMI_STS)

30.7.1.7 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.

30.7.2 SMBus Slave Interface

The PCH Secondary SMBus interface is accessed using the SMBus. The secondary SMBus logic will not generate or handle receiving the PEC byte and will only act as a Legacy Alerting Protocol device. The secondary interface allows the PCH to decode cycles, and allows an external microcontroller to perform specific actions.

Key features and capabilities include:

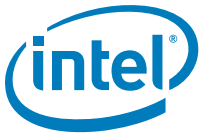
- Supports decode of three types of messages: Byte Write, Byte Read, and Host Notify.
- Receive Secondary Address register: This is the address that the PCH decodes. A default value is provided so that the secondary interface can be used without the processor having to program this register.
- Receive Secondary 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 secondary SMBus logic caused an interrupt or SMI# due to the reception of a message that matched the secondary address.
  - Bit 0 of the Secondary 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 secondary SMBus logic until either:

- 800 milliseconds after both: RTCRST# is high and RSMRST# is high, OR
- The PLTRST# de-asserts

If a primary 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 secondary logic's behavior is undefined. This is interpreted as an unexpected idle and should be avoided when performing management activities to the secondary logic.

Note: When an external microcontroller accesses the secondary SMBus Interface over the SMBus, a translation in the address is needed to accommodate the least significant bit used for read/write control. For example, if the PCH secondary address (RCV_SLVA) is left at 44h (default), the external micro controller would use an address of 88h/89h (write/read).



30.7.2.1 Format of Slave Write Cycle

The external primary performs Byte Write commands to the PCH Secondary SMBus I/F. The "Command" field (bits 11:18) indicate which register is being accessed. The Data field (bits 20:27) indicates the value that should be written to that register.

Table 30-5 has the values associated with the registers.

Table 30-5. Secondary Write Registers

Register	Function
0	Command Register. See Table 30-6 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 30-6. 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. ***Note:*** The SMB_WAK_STS bit will be set by this command, even if the system is already awake. The SMI handler should then clear this bit.
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 hard 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. ***Note:*** This command is only available in S0. All attempts to trigger a host reset without power cycle while the system is in Sx are dropped
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. **Note:** The command is supported in the following scenarios: • If the system is in Sx/M3or Sx/M3-PG, the command is supported. • If the system is in Sx/Moff, the command is supported if performed after a graceful Sx entry (i.e. if the platform was put to sleep or turned off via a write to the SLP_TYP/ SLP_EN fields by the OS or BIOS). Otherwise, the command is not supported.
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 de-assertion of the RSMRST# signal.
6	**WD RELOAD:** Reload watchdog timer.
7	Reserved

Table 30-6. Command Types (Sheet 2 of 2)

Command Type	Description
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.

30.7.2.2 Format of Read Command

The external primary performs Byte Read commands to the PCH secondary SMBus interface. The "Command" field (bits 18:11) indicate which register is being accessed. The Data field (bits 30:37) contains the value that should be read from that register.

Table 30-7. Secondary Read Cycle Format

Bit	Description	Driven By	Comment
1	Start	External Microcontroller	
2–8	Secondary Address - 7 bits	External Microcontroller	Must match value in Receive Secondary 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. See Table 30-8 for a list of implemented registers.
19	ACK	PCH	
20	Repeated Start	External Microcontroller	
21–27	Secondary Address - 7 bits	External Microcontroller	Must match value in Receive Secondary Address register
28	Read	External Microcontroller	Always 1
29	ACK	PCH	
30–37	Data Byte	PCH	Value depends on register being accessed. See Table 30-8 for a list of implemented registers.
38	NOT ACK	External Microcontroller	
39	Stop	External Microcontroller	

Table 30-8. Data Values for Secondary 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

Table 30-8. Data Values for Secondary Read Registers (Sheet 2 of 2)

Register	Bits	Description
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 = The Intruder Detect (INTRD_DET) bit is set. This indicates that the system cover has probably been opened.
	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 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 an 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
F	7:0	Year of the RTC
10h–FFh	7:0	Reserved

30.7.2.2.1 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 Secondary 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 Secondary 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 Secondary Address, it will assume that the protocol is followed, ignore bit 28, and proceed with the Secondary Read cycle.

Note: An external microcontroller must not attempt to access the PCH's Secondary SMBus logic until at least 1 second after both RTCRST# and RSMRST# are de-asserted (high).

Note: Until at least 1 second after both RTCRST# and RSMRST# are de-asserted (high).

30.7.2.3 Secondary Read of RTC Time Bytes

The PCH secondary SMBus interface allows external SMBus master to read the internal RTC's time byte registers.

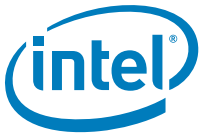
The RTC time bytes are internally latched by the PCH's hardware whenever RTC time is not changing and SMBus is idle. This ensures that the time byte delivered to the secondary read is always valid and it does not change when the read is still in progress on the bus. The RTC time will change whenever hardware update is in progress, or there is a software write to the RTC time bytes.

The PCH secondary SMBus interface only supports Byte Read operation. The external primary SMBus 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 primary SMBus 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 minutes 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.

30.7.2.4 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 primary 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 30-9 shows the Host Notify format.

Table 30-9. Host Notify Format

Bit	Description	Driven By	Comment
1	Start	External Primary	
8:2	SMB Host Address – 7 bits	External Primary	Always 0001_000
9	Write	External Primary	Always 0
10	ACK (or NACK)	PCH	PCH NACKs if HOST_NOTIFY_STS is 1
17:11	Device Address – 7 bits	External Primary	Indicates the address of the master; loaded into the Notify Device Address Register
18	Unused – Always 0	External Primary	7-bit-only address; this bit is inserted to complete the byte
19	ACK	PCH	
27:20	Data Byte Low – 8 bits	External Primary	Loaded into the Notify Data Low Byte Register
28	ACK	PCH	
36:29	Data Byte High – 8 bits	External Primary	Loaded into the Notify Data High Byte Register
37	ACK	PCH	
38	Stop	External Primary	

30.7.2.5 Format of Read Command

The external primary performs Byte Read commands to the PCH Secondary SMBus interface. The "Command" field (bits 18:11) indicate which register is being accessed. The Data field (bits 30:37) contains the value that should be read from that register.

Table 30-10. Secondary Read Cycle Format (Sheet 1 of 2)

Bit	Description	Driven By	Comment
1	Start	External Microcontroller	
2–8	Secondary Address - 7 bits	External Microcontroller	Must match value in Receive Secondary 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. See Table 30-11 for a list of implemented registers.
19	ACK	PCH	
20	Repeated Start	External Microcontroller	
21–27	Secondary Address - 7 bits	External Microcontroller	Must match value in Receive Secondary Address register
28	Read	External Microcontroller	Always 1
29	ACK	PCH	
30–37	Data Byte	PCH	Value depends on register being accessed. See Table 30-11 for a list of implemented registers.
38	NOT ACK	External Microcontroller	

Table 30-10. Secondary Read Cycle Format (Sheet 2 of 2)

Bit	Description	Driven By	Comment
39	Stop	External Microcontroller	

Table 30-11. Data Values for Secondary 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.** 1 = The Intruder Detect (INTRD_DET) bit is set. This indicates that the system cover has probably been opened.
	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 returns 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 an 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.

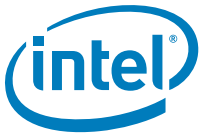


Table 30-11. Data Values for Secondary Read Registers (Sheet 2 of 2)

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 30-12. Enables for Secondary SMBus 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
Secondary Write to Wake/ SMI# Command	X	X	Wake generated when asleep. Secondary SMI# generated when awake (SMBUS_SMI_STS)
Secondary Write to SMLINK_SLAVE_SMI Command	X	X	Secondary 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

§ §

31 Serial Peripheral Interface (SPI)

31.1 Acronyms

Acronyms	Description
MISO	Master In Slave Out
MOSI	Master Out Slave In
SPI	Serial Peripheral Interface

31.2 References

None

31.3 Overview

The PCH provides one Serial Peripheral Interface (SPI). The interface implements 3 Chip Select signals (CS#), allowing up to two flash devices and one TPM device to be connected to the PCH. The CS0# and CS1# are used for flash devices and CS2# is dedicated to TPM.

The SPI interfaces support either 1.8V or 3.3V.

Note: The SPI interface covered in this chapter is for flash and TPM support only. This interface is distinct from other SPI described in this document such as the Generic SPI (GSPI).

31.4 Signal Description

Name	Type	Description
SPI0_CLK	O	**SPI Clock**: SPI clock signal for the common flash/TPM interface. Supports 17 MHz, 30 MHz and 48 MHz.
SPI0_CS0#	O	**SPI Chip Select 0**: Used to select the primary SPI Flash device. ***Note:*** This signal cannot be used for any other type of device than SPI Flash.
SPI0_CS1#	O	**SPI Chip Select 1**: Used to select an optional secondary SPI Flash device. ***Note:*** This signal cannot be used for any other type of device than SPI Flash.
SPI0_CS2#	O	**SPI Chip Select 2:** Used to select the TPM device if it is connected to the SPI interface; it cannot be used for any other type of device. ***Note:*** TPM can be configured through soft straps to operate over LPC or SPI, but no more than 1 TPM is allowed in the system.
SPI0_MOSI	I/O	**SPI 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 (SPI_IO0) to support the new Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes.

Name	Type	Description
SPI0_MISO	I/O	**SPI 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 (SPI_IO1) to support the new Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes.
SPI0_IO[3:2]	I/O	**SPI 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.

31.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
SPI0_MOSI	Pull-up	15K - 40K	
SPI0_MISO	Pull-up	15K - 40K	
SPI0_IO[2:3]	Pull-up	15K - 40K	

31.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
SPI0_CLK	Primary	Internal Pull-down (See Note 1)	Driven Low	Driven Low	Off
SPI0_CLK	Primary	Undriven (See Note 1)	Driven Low	Driven Low	Off
SPI0_MOSI	Primary	Internal Pull-up (See Note 1)	Driven Low	Driven Low	Off
SPI0_MISO	Primary	Internal Pull-up (See Note 1)	Internal Pull-up	Internal Pull-up	Off
SPI0_CS[2:0]#	Primary	Internal Pull-down (See Note 1)	Driven High	Driven High	Off
SPI0_CS[2:0]#	Primary	Undriven (See Note 1)	Driven High	Driven High	Off
SPI0_IO[3:2]	Primary	Internal Pull-up (See Note 1)	Internal Pull-up	Internal Pull-up	Off
Notes: 1. Pins are tri-stated prior to RSMRST# de-assertion.					

31.7 Functional Description

31.7.1 SPI for Flash

31.7.1.1 Overview

The PCH supports up to two SPI flash devices using two separate Chip Select pins. 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 using 3-byte addressing or 64 MB using 4-byte addressing.

The PCH SPI interface supports approximate frequencies of 17-MHz, 30-MHz, and 48-MHz. A flash device meeting 66-MHz timing is required for 48-MHz operation.

The SPI interface supports either 3.3V or 1.8V.

A SPI Flash device on Chip Select 0 (SPI_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 SPI has a third chip select SPI_CS2# for TPM support over SPI. TPM Bus will use SPI_CLK, SPI_MISO, SPI_MOSI and SPI_CS2# SPI signals.

Notes:

1. If Boot BIOS Strap ='00' then LPC is selected as the location for BIOS. BIOS may still be placed on LPC, but all platforms with the PCH require a SPI flash connected directly to the PCH's SPI bus with a valid descriptor connected to Chip Select 0 in order to boot.
2. When SPI is selected by the Boot BIOS Destination Strap and a SPI device is detected by the PCH, LPC-based BIOS flash is disabled.

31.7.1.2 SPI Supported Features

31.7.1.2.1 Descriptor Mode

Descriptor Mode is required for all SKUs of the PCH. Non-Descriptor Mode is not supported.

31.7.1.2.2 SPI Flash Regions

In Descriptor Mode the Flash is divided into five separate regions.

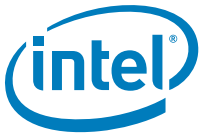
Table 31-1. SPI Flash Regions

Region	Content
0	Flash Descriptor
1	BIOS
2	Intel Management Engine
3	Gigabit Ethernet
4	Platform Data
8	EC

Only four primaries 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® ME 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

SPI 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® ME and BIOS regions. The Intel ME region contains firmware to support Intel Active Management Technology and other Intel ME capabilities.

Table 31-2. Region Size Versus Erase Granularity of Flash Components

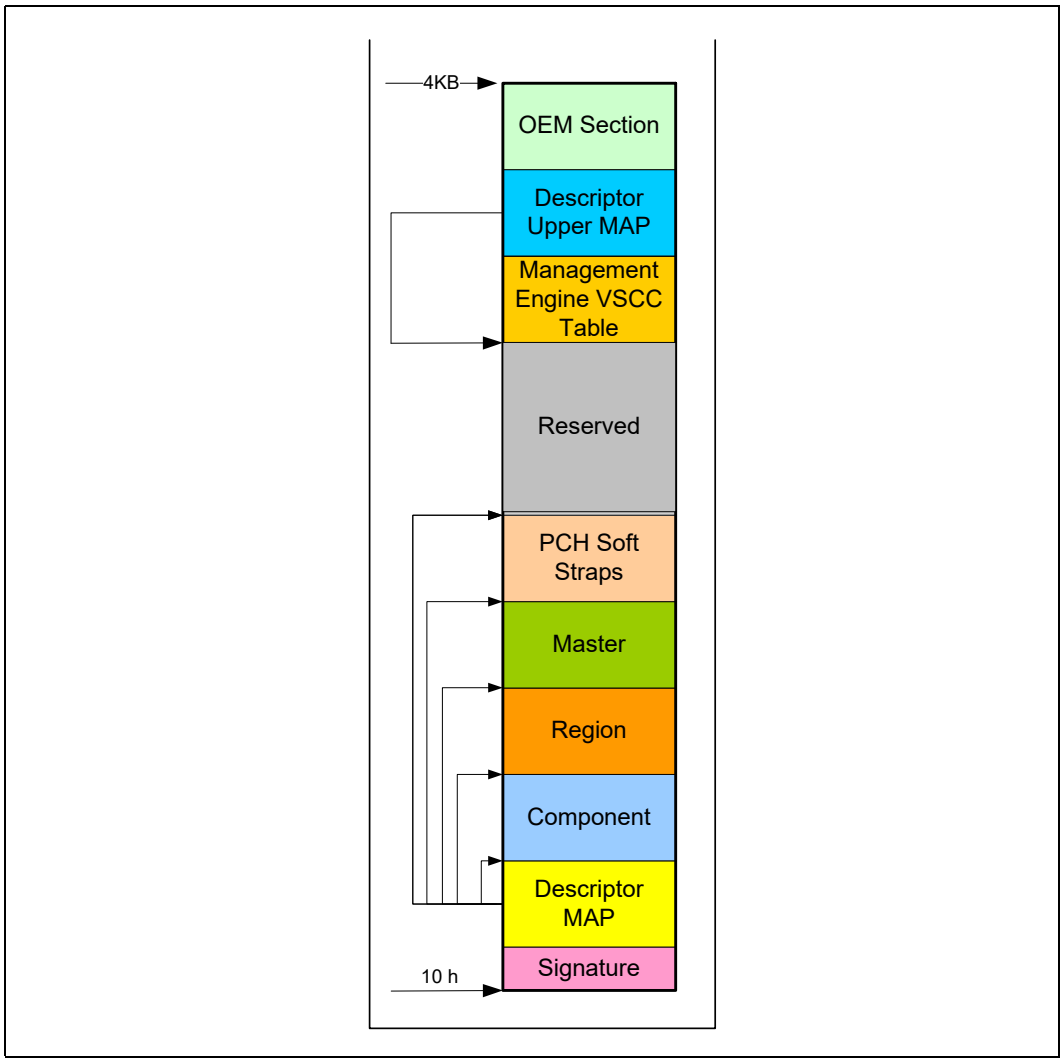
Region	Size with 4-KB Blocks	Size with 8-KB Blocks	Size with 64-KB Blocks
Descriptor	4 KB	8 KB	64 KB
GbE	8 KB	16 KB	128 KB
BIOS	Varies by Platform	Varies by Platform	Varies by Platform
Intel® ME	Varies by Platform	Varies by Platform	Varies by Platform
EC	Varies by Platform	Varies by Platform	Varies by Platform

31.7.1.3 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 SPI 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 31-1.

Figure 31-1. 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 SPI 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 primary region contains the security settings for the flash, granting read/write permissions for each region and identifying each primary by a requestor 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 SPI Flash supported by the NVM image.
- OEM Section is 256 bytes reserved at the top of the Flash Descriptor for use by OEM.

31.7.1.3.1 Descriptor Primary Region

The primary region defines read and write access setting for each region of the SPI device. The primary region recognizes four primaries: BIOS, Gigabit Ethernet, Management Engine, and EC. Each primary is only allowed to do direct reads of its primary regions.

Table 31-3. Region Access Control Table

Primary Read/Write Access				
Region	**Processor and BIOS**	**Intel® ME**	**GbE Controller**	**EC**
Descriptor	N/A	N/A	N/A	N/A
BIOS	Processor and BIOS can always read from and write to BIOS Region	Read/Write	Read/Write	Read/Write
Intel® Management Engine (CSME)	Read/Write	Intel® ME can always read from and write to Intel® ME Region	Read/Write	Read/Write
Gigabit Ethernet	Read/Write	Read/Write	GbE software can always read from and write to GbE region	Read/Write
Platform Data Region	N/A	N/A	N/A	N/A
EC	N/A	N/A	N/A	EC can always read from and write to EC region

31.7.1.4 Flash Access

There are two types of accesses: Direct Access and Program Register Accesses.

31.7.1.4.1 Direct Access

- Primaries 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.
- Primary's Host or Management Engine virtual read address is converted into the SPI 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 Primary 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 primary
  - Supports the same cache flush mechanism in ICH7 which includes Program Register Writes.

31.7.1.4.2 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 primaries can access the registers

Note: Processor running Gigabit Ethernet software can access Gigabit Ethernet registers:

- Primaries are only allowed to read or write those regions they have read/write permission
- Using the Flash Region Access Permissions, one primary can give another primary read/write permissions to their area
- Using the five Protected Range registers, each primary can add separate read/write protection above that granted in the Flash Descriptor for their own accesses
  - Example: BIOS may want to protect different regions of BIOS from being erased
  - Ranges can extend across region boundaries

31.7.2 SPI Support for TPM

The PCH's SPI flash controller supports a discrete TPM on the platform via its dedicated SPI0_CS#2 signal. The platform must have no more than 1 TPM.

SPI controller supports accesses to SPI TPM at approximately 17 MHz, 30 MHz or 48 MHz, depending on the PCH soft strap. 17 MHz is the reset default; a valid PCH soft strap setting overrides the requirement for the 17 MHz. SPI TPM device must support a clock of 17 MHz, and thus should handle 15-20 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 SPI controller.

Note that the SPI controller is configurable to prevent TPM access when the descriptor is invalid (or no flash is attached).

§ §

32 Super Speed Inter-Chip

32.1 Acronyms

Acronyms	Description
SSIC	Super Speed Inter-Chip

32.2 References

Specification	Location
Inter-Chip Supplement to the USB Revision 3.0 Specifications	www.usb.org
Universal Serial Bus Revision 3.0 Specification including ECNs	www.usb.org
High-Speed Inter-Chip USB Electrical Specification including ECNs	www.usb.org
MIPI* Alliance Specification for M-PHYSM	www.usb.org
MIPI M-PHY Conformance Test Suite	www.usb.org

32.3 Overview

SSIC is an inter-chip interface that uses USB 3.0 specifications with the transfer speeds up to 1.45 Gbps. There is only one USB 3.0 multiplexed port available to use as an SSIC port. Use SSIC port 1.

Note: PCH-U/Y only supports SSIC x1 Gear 1 Rate B up to 1.45 Gbps.

32.4 Signal Description

Name	Type	Description
USB3_2_TXN/ **SSIC_TXN,** USB3_2_TXP/ **SSIC_TXP**	O	**SSIC Differential Transmit Pair 1:** These are muxed with USB 3.0 Port #2 outbound differential signals.
USB3_2_RXN/ **SSIC_RXN,** USB3_2_RXP/ **SSIC_RXP**	I	**SSIC Differential Receive Pair 1:** These are multiplexed with USB 3.0 Port #2 inbound differential signals.

32.5 Integrated Pull-Ups and Pull-Downs

None

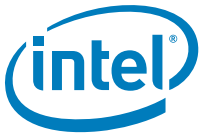
32.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
SSIC_TXN **SSIC_TXP**	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SSIC_RXN **SSIC_RXP**	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF

32.7 Functional Description

SSIC uses the MIPI* M-PHY specification as the physical layer of the interconnect to meet the requirements of the embedded inter-chip interfaces. The M-PHY specification describes a serial physical layer technology with high-bandwidth capability, which is specifically developed for the mobile applications to obtain low pin count combined with good power efficiency. SSIC on PCH supports x1 configuration. One USB 3.0 multiplexed port is available for the SSIC interface.

§ §



33 Testability

33.1 JTAG

33.1.1 Acronyms

Acronyms	Description
BSDL	Boundary Scan Description Language
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input/Output
I/OD	Input/Output Open Drain
JTAG	Joint Test Action Group

33.1.2 References

Specification	Location
IEEE Standard Test Access Port and Boundary Scan Architecture	http://standards.ieee.org/findstds/standard/1149.1-2013.html

33.1.3 Overview

This section contains information regarding the PCH testability signal that provides access to JTAG, run control, system control, and observation resources. PCH 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)

33.1.4 Signal Description

Name	Type	Description
PCH_JTAG_TCK	I/O	**Test Clock Input (TCK):** The test clock input provides the clock for the JTAG test logic.
PCH_JTAG_TMS	I/OD	**Test Mode Select (TMS):** The signal is decoded by the Test Access Port (TAP) controller to control test operations.
PCH_JTAG_TDI	I/OD	**Test Data Input (TDI):** Serial test instructions and data are received by the test logic at TDI.
PCH_JTAG_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.
JTAGX	I/O	This pin is used to support merged debug port topologies.
ITP_PMODE	O	This signal is used to transmit processor and PCH power/reset information to the ITP Debugger.
PCH_TRST#	O	JTAG output from DCI to CPU

33.1.5 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
PCH_JTAG_TCK	Primary	Internal PD	Internal PD	Internal PD	Off
PCH_JTAG_TMS	Primary	Internal PU	Internal PU	Internal PU	Off
PCH_JTAG_TDI	Primary	Internal PU	Internal PU	Internal PU	Off
PCH_JTAG_TDO	Primary	Undriven	Undriven	Undriven	Off
JTAGX[1]	Primary	Internal PU (as TDO)/Internal PD (as TCK)	Internal PU/ Internal PD	Internal PU/ Internal PD	Off
ITP_PMODE[2]	Primary	Internal PU	Internal PU	Internal PU	Off
PCH_TRST#	Primary	Internal PD	Internal PD	Internal PD	Off

Notes:

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. This pin is connected to HOOK[6] on the merged debug topology.

33.2 Intel® Trace Hub (Intel® TH)

33.2.1 Overview

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:

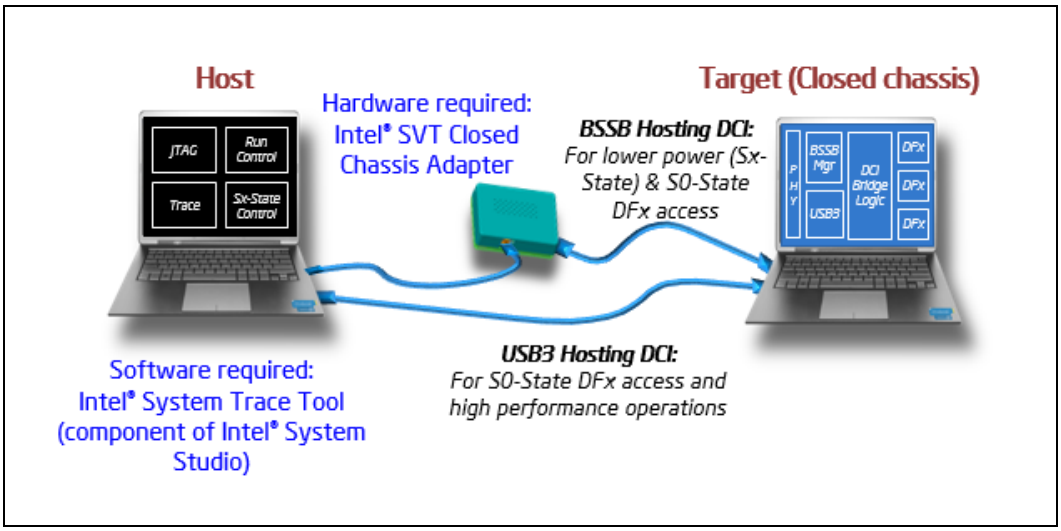
- Direct Connect Interface (DCI)
  - — BSSB Hosting DCI
  - — USB3 Hosting DCI
- System Memory

There are multiple trace sources planned to be supported in the platform:

- BIOS
- CSME
- AET (Architecture Event Trace)
- PCH Power Management Event Trace
- PCH Hardware Signals
- Windows* ETW (for driver or application)

33.2.2 Platform Setup

Figure 33-1. Platform Setup with Intel® Trace Hub



33.3 Direct Connect Interface (DCI)

Direct Connect Interface (DCI) is a new debug transport technology to enable closed chassis debug through any of USB3 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-based debugger (kernel level debugger) are mutually exclusive.

There are two types of DCI hosting interfaces in the platform:

- BSSB Hosting DCI
- USB3 Hosting DCI

Supported capabilities in DCI are:

- Closed Chassis Debug at S0 & Sx State
- JTAG Access & Run Control (Probe Mode)
- System Tracing with Intel® Trace Hub

Debug host softwares that support DCI are:

- Intel® ITP II Platform Debug Toolkit (PDT)
- Intel® System Studio (ISS)

33.3.1 Boundary Scan Side Band (BSSB) Hosting DCI

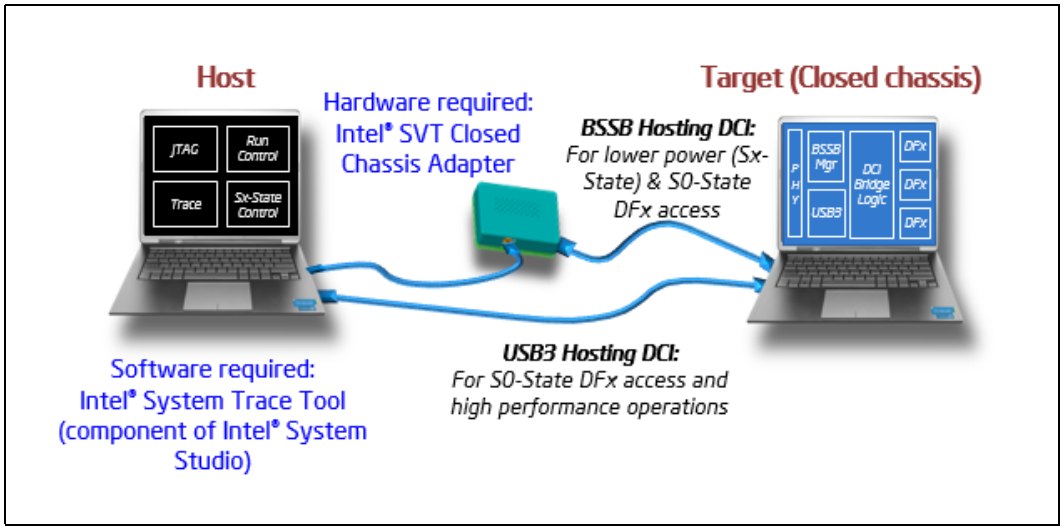
BSSB was developed to provide an alternate path to convey controls and data to or from the embedded logic by connecting physically to the target through a USB3 port. BSSB provides an alternate side band path around the USB3 controller, so that the embedded logic can be accessed, even when the USB controller is not alive (such as in low-power states), or is malfunctioning. This path does not rely on USB 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 developed by Intel called Intel® SVT Closed Chassis Adapter (CCA). It is a simple data transformation device. This adapter generates a BSSB 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 the moral equivalent of the USB link layer. This path also uses native DCI packet protocol instead of USB protocol.

33.3.2 USB3 Hosting DCI

It relies on Debug Class Devices (DbC) which is composed 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 device for debug purpose. This path uses the USB packet protocol layer, USB link layer flow control and USB3 physical layer at 5 GHz.

33.3.3 Platform Setup

Figure 33-2. Platform Setup with DCI Connection



§ §

34 Intel® Serial I/O Universal Asynchronous Receiver/ Transmitter (UART) Controllers

34.1 Acronyms

Acronyms	Description
DMA	Direct Memory Access
UART	Universal Asynchronous Receiver/Transmitter

34.2 References

None

34.3 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

Note: Bluetooth® devices are not supported on the PCH UART interfaces.

34.4 Signal Description

Name	Type	Description
UART0_RXD/ GPP_C8	I	**UART 0 Receive Data**
UART0_TXD/ GPP_C9	O	**UART 0 Transmit Data**
UART0_RTS#/ GPP_C10	O	**UART 0 Request to Send**
UART0_CTS#/ GPP_C11	I	**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**

Name	Type	Description
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**

34.5 Integrated Pull-Ups and Pull-Downs

None

34.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	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

34.7 Functional Description

34.7.1 Features

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))

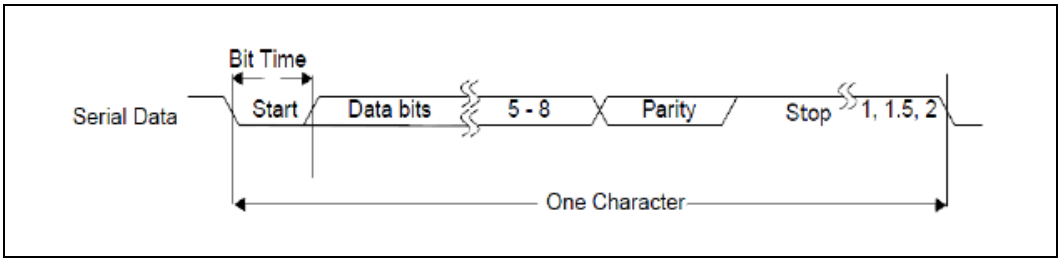
Notes:

1. SIR mode is not supported.
2. Dual clock is not supported.
3. External read enable signal for RAM wake up when using external RAMs is not supported.

34.7.2 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.

Figure 34-1. UART Serial Protocol



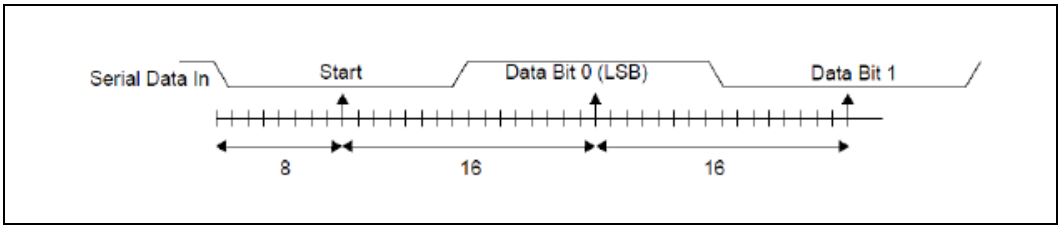
The UART Host Controller Line Control Register (LCR) is used to control the serial character characteristics. The individual bits of the data word are sent after the Start bit, starting with the least significant bit (LSB). These are followed by the optional parity bit, followed by the Stop bit(s), which 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 34-2. UART Receiver Serial Data Sample Points



34.7.3 16550 8-bit Addressing - Debug Driver Compatibility

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 control of this mode is via the GEN_REGGRW7 register (UART Additional Registers, offset 0x618h). Refer to register section for the description of these bits.

Note: The UART 16550 8-bit Legacy mode only operates with PIO transactions. DMA transactions are not supported in this mode.

34.7.4 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.

34.7.4.1 DMA Transfer and Setup Modes

The DMA can operate in the following modes:

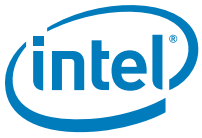
1. Memory to peripheral transfers. This mode requires that the peripheral control the flow of the data to itself.
2. Peripheral to memory transfer. This mode requires that the peripheral control the flow of the data from itself.

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.

34.7.4.2 Channel Control

- The source transfer width and destination transfer width are programmed. 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.

34.7.5 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.

34.7.6 Power Management

34.7.6.1 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.

34.7.6.2 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 represents 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.

34.7.7 Interrupts

UART interface has an interrupt line which is used to notify the driver that service is required.

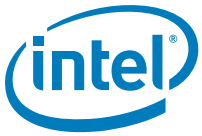
When an interrupt occurs, the device driver needs to read both the host controller and DMA interrupt status registers to identify the interrupt source. Clearing the interrupt is done with the corresponding interrupt register in the host controller or DMA.

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

34.7.8 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.

§ §



35 Universal Serial Bus (USB)

35.1 Acronyms

Acronyms	Description
xHCI	eXtensible Host Controller Interface

35.2 References

Specification	Location
USB 3.0 Specification	www.usb.org
USB 2.0 Specification	www.usb.org

35.3 Overview

The PCH implements an xHCI USB controller which provides support for up to 10 USB 2.0 signal pairs and 6 SuperSpeed USB 3.0 signal pairs. The xHCI controller supports wake up from sleep states S1-S4. The xHCI USB controller supports up to 64 devices and 128 endpoints.

Note: Each walk-up USB 3.0 capable port must have USB 3.0 signaling and USB 2.0 signaling.

Note: EHCI is no longer supported in PCH.

35.4 Signal Description

Name	Type	Description
USB3_1_RXN, USB3_1_RXP	I	**USB 3.0 Differential Receive Pair 1:** These are USB 3.0-based high-speed differential signals for Port #1 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). This port also supports Dual Role Capability for USB On The Go.
USB3_1_TXN, USB3_1_TXP	O	**USB 3.0 Differential Transmit Pair 1:** These are USB 3.0-based high-speed differential signals for Port #1 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). This port also supports Dual Role Capability for USB On The Go.
USB3_2_RXN / SSIC_RXN, **USB3_2_RXP**/ SSIC_RXP	I	**USB 3.0 Differential Receive Pair 2:** These are USB 3.0-based high-speed differential signals for Port #2 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). This port can also be used as a SSIC.
USB3_2_TXN / SSIC_TXN, **USB3_2_TXP**/ SSIC_TXP	O	**USB 3.0 Differential Transmit Pair 2:** These are USB 3.0-based high-speed differential signals for Port #2 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). This port can also be used as a SSIC.
USB3_3_RXN, USB3_3_RXP	I	**USB 3.0 Differential Receive Pair 3:** These are USB 3.0-based high-speed differential signals for Port #3 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent).
USB3_3_TXN, USB3_3_TXP	O	**USB 3.0 Differential Transmit Pair 3:** These are USB 3.0-based high-speed differential signals for Port #3 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent).

Name	Type	Description
USB3_4_RXN, USB3_4_RXP	I	**USB 3.0 Differential Receive Pair 4:** These are USB 3.0-based high-speed differential signals for Port #4 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent).
USB3_4_TXN, USB3_4_TXP	O	**USB 3.0 Differential Transmit Pair 4:** These are USB 3.0-based high-speed differential signals for Port #4 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent).
USB3_5_RXN/ PCIE1_RXN **USB3_5_RXP**/ PCIE1_RXP	I	**USB 3.0 Differential Receive Pair 5:** These are USB 3.0-based high-speed differential signals for Port #5 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 1.
USB3_5_TXN/ PCIE1_TXN **USB3_5_TXP/** PCIE1_TXP	O	**USB 3.0 Differential Transmit Pair 5:** These are USB 3.0-based high-speed differential signals for Port #5 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 1.
USB3_6_RXN/ PCIE2_RXN **USB3_6_RXP/** PCIE2_RXP	I	**USB 3.0 Differential Receive Pair 6:** These are USB 3.0-based high-speed differential signals for Port #6 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 2.
USB3_6_TXN/ PCIE2_TXN **USB3_6_TXP/** PCIE2_TXP	O	**USB 3.0 Differential Transmit Pair 6:** These are USB 3.0-based high-speed differential signals for Port #6 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). ***Note:*** Use FITC to set the soft straps that select this port as PCIe* Port 2.
USB2P_1, USB2N_1	I/O	**USB 2.0 Port 1 Transmit/Receive Differential Pair 1:** This USB 2.0 signal pair is routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. This port also supports Dual Role Capability for USB On The Go.
USB2P_2, USB2N_2	I/O	**USB 2.0 Port 2 Transmit/Receive Differential Pair 2:** This USB 2.0 signal pair is routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC.
USB2P_3, USB2N_3	I/O	**USB 2.0 Port 3Transmit/Receive Differential Pair 3:** This USB 2.0 signal pair is routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC.
USB2P_4, USB2N_4	I/O	**USB 2.0 Port 4 Transmit/Receive Differential Pair 4:** This USB 2.0 signal pair is routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC.
USB2P_5, USB2N_5	I/O	**USB 2.0 Port 5 Transmit/Receive Differential Pair 5:** This USB 2.0 signal pair is routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC.
USB2P_6, USB2N_6	I/O	**USB 2.0 Port 6 Transmit/Receive Differential Pair 6:** This USB 2.0 signal pair is routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC.
USB2P_7, USB2N_7	I/O	**USB 2.0 Port 7 Transmit/Receive Differential Pair 7:** This USB 2.0 signal pair is routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC.
USB2P_8, USB2N_8	I/O	**USB 2.0 Port 8 Transmit/Receive Differential Pair 8:** This USB 2.0 signal pair is routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC.
USB2P_9, USB2N_9	I/O	**USB 2.0 Port 9 Transmit/Receive Differential Pair 9:** This USB 2.0 signal pair is routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC.
USB2P_10, USB2N_10	I/O	**USB 2.0 Port 10 Transmit/Receive Differential Pair 10:** This USB 2.0 signal pair is routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC.

Name	Type	Description
USB2_OC0#/ GPP_E9	I	**Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred.
USB2_OC1# / GPP_E10	I	**Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred.
USB2_OC2#/ GPP_E11	I	**Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred.
USB2_OC3#/ GPP_E12	I	**Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred.
USB2_VBUSSENSE	I	VBUS Sense for USB On-The-Go. Refer to OTG 2.0 specification for the sensing threshold voltage spec.
USB2_ID	I	ID detects for USB On The Go.
USB2_COMP	I	USB Resistor Bias, analog connection points for an external resistor to ground.

35.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
USB2N_[10:1]	Internal Pull-down	14.25–24.8K	1
USB2P_[10:1]	Internal Pull-down	14.25–24.8K	1
USB2_ID	Internal Weak Pull-up		If this signal is not in use, then it should have 1k PD to ground
Note: 1. Series resistors (45 ohm ±10%)			

35.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately After Reset	S3/S4/S5	Deep Sx
USB3_[6:1]_RXN **USB3_[6:1]_RXP**	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
USB3_[6:1]_TXN **USB3_[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
USB2_OC0#	Primary	Undriven	Undriven	Undriven	OFF
USB2_OC1#	Primary	Undriven	Undriven	Undriven	OFF
USB2_OC2#	Primary	Undriven	Undriven	Undriven	OFF
USB2_OC3#	Primary	Undriven	Undriven	Undriven	OFF
USB2_VBUSSENSE	Primary	Undriven	Undriven	Undriven	OFF
USB2_ID[1]	Primary	Internal Pull-up	Undriven/ Internal Pull-up	Undriven/ Internal Pull-up	OFF
USB2_COMP	Primary	Undriven	Undriven	Undriven	OFF

Notes:

1. The USB2 ID pin is pulled-up internally.

35.7 Functional Description

35.7.1 eXtensible Host Controller Interface (xHCI) Controller (D20:F0)

The PCH contains an eXtensible Host Controller Interface (xHCI) host controller which supports up to 10–USB 2.0 ports and up to6–USB 3.0 ports with board routing, ACPI table and BIOS considerations. This controller allows data transfers of up to 5Gb/s. The controller supports SuperSpeed (SS), High-Speed (HS), Full-Speed (FS) and Low-Speed (LS) traffic on the bus. The xHCI controller supports USB Debug port on all USB 3.0-capable ports. The xHCI also supports USB Attached SCIS Protocol (UASP).

The PCH also supports Dual Role Capability. The USB Host Controller can now be paired with a standalone USB device to provide dual role functionality. The USB subsystem incorporates a USB 3.0 device controller. This controller is instantiated as a separate PCI function and shares USB 2.0 port 1 and USB 3.0 port 1. The PCH USB implementation is compliant to the Device specification and supports host/device control through ID pin. The ID pin is an input micro AB connector and signifies the type of agent connected to the port.

35.7.1.1 USB Dual Role Support

The Device controller shares USB 3.0 port #1 and USB 2.0 port #1 with the host controller, with ownership of the port being decided by the ID pin. A 1 on the ID pin signifies that the port is to be mapped to the device controller. A 0 signifies that the port is to be mapped to the host controller. While the port is mapped to the device controller the host controller Rx detection must always indicate a disconnected port. Only PCH-U/Y SKUs support Dual Role functionality.

§ §

36 Camera Serial Interface

36.1 Acronyms

Acronyms	Description
CSI-2	Camera Serial Interface 2

36.2 References

Specification	Location
MIPI* Specification for CSI-2 1.01	http://www.mipi.org/specifications/camera-interface
MIPI* Specification for D-PHY 1.1	

36.3 Overview

The Camera I/O Controller provides a native/integrated interconnect to camera sensors, compliant with MIPI* CSI-2 protocol. A total of 12 lanes are available for the camera interface supporting up to 4 sensors. This interface is only available on PCH-U/Y SKUs.

36.4 Signal Description

Name	Type	Description
CSI2_DP[11:0] **CSI2_DN[11:0]**	I	MIPI CSI-2 Data
CSI2_CLKP[3:0] **CSI2_CLKN[3:0]**	I	MIPI CSI-2 Clock
FLASHTRIG/ GPP_D4	O	GPIO camera I/O flash trigger
CSI2_COMP	I	MIPI CSI-2 Compensation (100 Ohm+/- 1% pull-down to ground)

36.5 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
FLASHTRIG/GPP_D4	None	None	
CSI2_COMP	None	None	

36.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
CSI2_DP[11:0] **CSI2_DN[11:0]**	Primary	Undriven	Undriven	Undriven	Off

Signal Name	Power Plane	During Reset	Immediately after Reset	S3/S4/S5	Deep Sx
CSI2_CLKP[3:0] **CSI2_CLKN[3:0]**	Primary	Undriven	Undriven	Undriven	Off
FLASHTRIG/ GPP_D4/	Primary	Undriven	Undriven	Undriven	Off
CSI2_COMP	Primary	Undriven	Undriven	Undriven	Off

36.7 Functional Description

There are 12 differential MIPI CSI-2 compliant lanes available, each capable of bandwidth up to 1.5 Gbps. Minimum 1 lane camera sensor can be used. CSI-2 interface supports up to 4 camera subsystems in various configurations. Data is transmitted from the camera sensor to the PCH using unidirectional CSI-2 data lanes. Camera Control Interface utilizes I^2C for the bidirectional communication. Camera I/O flash trigger GPIO output is used allowing the Camera I/O controller direct control of when the camera flash asserts. Some additional GPIOs may be needed for the camera LED, strobe, and so on.

§ §

37 embedded Multimedia Card (eMMC*)

37.1 Acronyms

Acronyms	Description
eMMC*	Embedded MultiMedia Card

37.2 References

Specification	Location
eMMC* Jedec Standard	http://www.jedec.org

37.3 Overview

The eMMC* 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. It supports 1.8V only devices.

37.4 Signal Description

Name	Type	Description
EMMC_CMD / GPP_F12	I/O	eMMC* Command/Response
EMMC_DATA0 / GPP_F13	I/O	eMMC* Data
EMMC_DATA1 / GPP_F14	I/O	eMMC* Data
EMMC_DATA2 / GPP_F15	I/O	eMMC* Data
EMMC_DATA3 / GPP_F16	I/O	eMMC* Data
EMMC_DATA4 / GPP_F17	I/O	eMMC* Data
EMMC_DATA5 / GPP_F18	I/O	eMMC* Data
EMMC_DATA6 / GPP_F19	I/O	eMMC* Data
EMMC_DATA7 / GPP_F20	I/O	eMMC* Data
EMMC_RCLK/ GPP_F21	I	eMMC* Receive Clock
EMMC_CLK/ GPP_F22	O	eMMC* Clock
EMMC_RCOMP	I/O	eMMC* compensation (200 Ohm +/- 1% pull down to ground)

37.5 Integrated Pull-Ups and Pull-Downs

None

37.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	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

37.7 Functional Description

The Controller handles eMMC* Protocol at transmission, packing data, adding cyclic redundancy check (CRC), start/end bit, and checking for transaction format correctness. It supports eMMC* 5.0 with the maximum data rate of 384.6 MB/s and eMMC* 4.5 with maximum data rate of 192.2 MB/s. It supports the data transfer in 1-bit, 4-bit and 8-bit modes. Secure and non-secure boot is not supported.

The eMMC* main use case is to connect an on board external storage device. The specification is defined in the JEDEC standard that can be reviewed for the additional information.

The following chart maps the working modes of eMMC*.

Table 37-1. eMMC* Working Modes

eMMC* Mode	Data Rate	Clock Freq	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

§ §

38 Secure Digital eXtended Capacity (SDXC)

38.1 Acronyms

Acronyms	Description
SDXC	Secure Digital eXtended Capacity

38.2 References

Specification	Location
SDXC Specifications	http://www.sdcard.org

38.3 Overview

The SDXC controller is to connect to an external detachable storage device. It supports SDXC specification version 3.01.

38.4 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	SD Card Detect	3.3V or 1.8V
SD_CLK/GPP_G6	O	SD clock	3.3V or 1.8V
SD_WP/GPP_G7	I	SD Card write protect	3.3V or 1.8V
SD_1P8_SEL/ GPP_A16	O	SD Card 1.8v Drive Voltage Select (Use 3.3v when low)	3.3V or 1.8V
SD_PWR_EN_# / ISH_GP7 / GPP_A17	O	SD card power enable	3.3V or 1.8V
SD_RCOMP	I/O	External reference resistor (200 Ohm+/- 1% pull down to ground)	N/A

38.5 Integrated Pull-Ups and Pull-Downs

None

38.6 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	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_1P8_SEL	Primary	Undriven	Undriven	Undriven	Off
SD_PWR_EN_#	Primary	Undriven	Undriven	Undriven	Off
SD_RCOMP	Primary	Undriven	Undriven	Undriven	Off

38.7 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 signalling is supported. Additional information can be obtained from the SDXC 3.0 specification.

The following chart maps the working modes of SD Card.

Table 38-1. SD Working Modes

SD Card Mode	Data Rate	Clock Frequency	Maximum Data Throughput
Default Speed/SDR12[1]	Single	0 – 25 MHz	12.5 MB/s
High Speed/SDR25[2]	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

§ §