

Intel® 700 Series Chipset Family On-Package Platform Controller Hub (PCH)

Datasheet, Volume 1 of 2

Rev. 001

January 2023

Doc. No.: 763122, Rev.: 001



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.

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 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.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com].

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

Copyright © 2023, Intel Corporation. All rights reserved.

Contents

Revision History....12

1.0 Introduction....13

1.1 Overview....13
1.2 PCH SKUs....14
1.3 Flexible High Speed I/O....15
1.3.1 PCH-P....16

2.0 PCH Controller Device IDs....17

2.1 Device and Revision ID....17

3.0 Memory Mapping....19

3.1 Functional Description....19
3.1.1 PCI Devices and Functions....19
3.1.2 Fixed I/O Address Ranges....19
3.1.3 Variable I/O Decode Ranges....22
3.2 Memory Map....23
3.2.1 Boot Block Update Scheme....25

4.0 System Management....27

4.1 Theory of Operation....27
4.1.1 Handling an Intruder....27
4.1.2 TCO Modes....28

5.0 High Precision Event Timer (HPET)....31

5.1 Feature Overview....31
5.1.1 Timer Accuracy....31
5.1.2 Timer Off-load....31
5.1.3 Interrupt Mapping....33
5.1.4 Periodic Versus Non-Periodic Modes....34
5.1.5 Enabling the Timers....36
5.1.6 Interrupt Levels....36

6.0 PCH Thermal Sensor....37

6.1 Modes of Operation....37
6.2 Temperature Trip Point....37
6.3 Thermal Sensor Accuracy ($T_{accuracy}$)....37
6.4 Thermal Reporting to an EC....38
6.5 Thermal Trip Signal (PCHHOT#)....38

7.0 Power Delivery....39

7.1 Power and Ground Signals....39

8.0 Pin Straps....40

9.0 Electrical and Thermal Characteristics....43

10.0 8254 Timers....44

10.1 Timer Programming....44
10.2 Reading from the Interval Timer....45

11.0 Audio Voice and Speech..... 47
11.1 Feature Overview.....47
11.1.1 Intel® High Definition Audio (Intel® HD Audio) Controller Capabilities.....48
11.1.2 Audio DSP Capabilities..... 48
11.1.3 Intel® High Definition Audio Interface Capabilities.....49
11.1.4 Direct Attached Digital Microphone (PDM) Interface..... 49
11.1.5 USB Audio Offload Support..... 50
11.1.6 Intel® Display Audio Interface..... 50
11.1.7 MIPI® SoundWire* Interface.....50
11.2 Signal Description..... 51
11.3 Integrated Pull-Ups and Pull-Downs.....54
11.4 I/O Signal Planes and States..... 54

12.0 Controller Link..... 56
12.1 Signal Description..... 56
12.2 Integrated Pull-Ups and Pull-Downs.....56
12.3 I/O Signal Planes and States..... 56
12.4 External CL_RST# Pin Driven/Open-drained Mode Support..... 57

13.0 Processor Sideband Signals..... 58
13.1 Functional Description.....58
13.2 Signal Description..... 58
13.3 Integrated Pull-Ups and Pull-Downs.....58
13.4 I/O Signal Planes and States..... 59

14.0 Digital Display Signals..... 60
14.1 Signal Description..... 60
14.2 Embedded DisplayPort* (eDP*) Backlight Control Signals..... 61
14.3 Integrated Pull-Ups and Pull-Downs.....62
14.4 I/O Signal Planes and States..... 62

15.0 Enhanced Serial Peripheral Interface (eSPI).....64
15.1 Functional Description.....64
15.1.1 Operating Frequency..... 65
15.1.2 Protocols 65
15.1.3 WAIT States from eSPI Slave.....65
15.1.4 In-Band Link Reset.....66
15.1.5 Slave Discovery 66
15.1.6 Flash Sharing Mode..... 66
15.1.7 PECI Over eSPI.....66
15.1.8 Multiple OOB Master.....66
15.1.9 Channels and Supported Transactions..... 67
15.2 Signal Description..... 72
15.3 Integrated Pull-Ups and Pull-Downs.....73
15.4 I/O Signal Planes and States..... 73

16.0 General Purpose Input and Output..... 74
16.1 Signal Description..... 74
16.2 Functional Description.....74
16.2.1 Configurable GPIO Voltage.....75
16.2.2 GPIO Buffer Impedance Compensation.....75
16.2.3 Interrupt / IRQ via GPIO Requirement.....75

16.2.4 Programmable Hardware Debouncer..... 75
16.2.5 Integrated Pull-ups and Pull-downs..... 76
16.2.6 SCI / SMI# and NMI..... 76
16.2.7 Timed GPIO..... 76
16.2.8 GPIO Blink (BK) and Serial Blink (SBK)..... 77
16.2.9 GPIO Ownership..... 77
16.2.10 Native Function and TERM Bit Setting..... 77

17.0 Intel® Serial I/O Inter-Integrated Circuit (I^2C) Controllers..... 78
17.1 Functional Description..... 79
17.1.1 Protocols Overview..... 79
17.1.2 DMA Controller..... 80
17.1.3 Reset..... 81
17.1.4 Power Management..... 81
17.1.5 Interrupts..... 82
17.1.6 Error Handling..... 82
17.1.7 Programmable SDA Hold Time..... 82
17.2 Signal Description..... 82
17.3 Integrated Pull-Ups and Pull-Downs..... 83
17.4 I/O Signal Planes and States..... 84

18.0 Gigabit Ethernet Controller..... 85
18.1 Functional Description..... 85
18.1.1 GbE PCI Express* Bus Interface..... 87
18.1.2 Error Events and Error Reporting..... 88
18.1.3 Ethernet Interface..... 88
18.1.4 PCI Power Management..... 88
18.2 Signal Description..... 90
18.3 Integrated Pull-Ups and Pull-Downs..... 91
18.4 I/O Signal Planes and States..... 92

19.0 Integrated Sensor Hub (ISH)..... 93
19.1 Functional Description..... 93
19.1.1 ISH Micro-Controller..... 93
19.1.2 SRAM..... 93
19.1.3 PCI Host Interface..... 94
19.1.4 Power Domains and Management..... 94
19.1.5 ISH IPC..... 94
19.1.6 ISH Interrupt Handling via IOAPIC (Interrupt Controller)..... 95
19.2 Feature Overview..... 95
19.2.1 ISH I^2C Controllers..... 96
19.2.2 ISH UART Controller..... 96
19.2.3 ISH GSPI Controller..... 96
19.2.4 ISH GPIOs..... 96
19.3 Signal Description 97
19.4 Integrated Pull-Ups and Pull-Downs..... 98
19.5 I/O Signal Planes and States..... 98

20.0 PCH and System Clocks..... 99
20.1 ICC..... 99
20.1.1 Signal Descriptions..... 99
20.2 I/O Signal Pin States..... 100

21.0 PCI Express* (PCIe*)..... 102
21.1 Functional Description..... 102
21.1.1 Interrupt Generation..... 102
21.1.2 PCI Express* Power Management..... 103
21.1.3 Dynamic Link Throttling..... 104
21.1.4 Port 8xh Decode..... 105
21.1.5 Separate Reference Clock with Independent SSC (SRIS)..... 105
21.1.6 Advanced Error Reporting..... 105
21.1.7 Single - Root I/O Virtualization (SR - IOV)..... 105
21.1.8 SERR# Generation..... 106
21.1.9 Hot - Plug..... 106
21.1.10 PCI Express* Lane Polarity Inversion..... 107
21.1.11 Precision Time Measurement (PTM)..... 107
21.2 Signal Description..... 107
21.3 I/O Signal Planes and States..... 109
21.4 PCI Express* Port Support Feature Details..... 109

22.0 Power Management..... 112
22.1 Functional Description..... 112
22.1.1 Features..... 113
22.1.2 Power Management States..... 113
22.1.3 PCH and System Power States..... 113
22.1.4 SMI#/SCI Generation..... 116
22.1.5 C-States..... 118
22.1.6 Sleep States..... 118
22.1.7 Event Input Signals and Their Usage..... 123
22.1.8 ALT Access Mode..... 128
22.1.9 System Power Supplies, Planes, and Signals..... 129
22.1.10 Legacy Power Management Theory of Operation..... 134
22.1.11 Reset Behavior..... 135
22.2 Signal Description..... 137
22.3 Integrated Pull-Ups and Pull-Downs..... 140
22.4 I/O Signal Planes and States..... 140

23.0 Real Time Clock (RTC)..... 143
23.1 Signal Description..... 143
23.2 I/O Signal Planes and States..... 144

24.0 Serial ATA (SATA)..... 145
24.1 Functional Description..... 145
24.1.1 SATA 6 Gb/s Support..... 145
24.1.2 SATA Feature Support..... 146
24.1.3 Hot - Plug Operation..... 146
24.1.4 Intel® Rapid Storage Technology (Intel® RST)..... 146
24.1.5 Power Management Operation..... 147
24.1.6 SATA Device Presence..... 149
24.1.7 SATA LED..... 150
24.1.8 Advanced Host Controller Interface (AHCI) Operation..... 150
24.1.9 Enclosure Management (SGPIO Signals)..... 151
24.2 Signals Description..... 153
24.3 Integrated Pull-Ups and Pull-Downs..... 154

24.4 I/O Signal Planes and States....154

25.0 System Management Interface and SMLink....155

25.1 Functional Description....155
25.1.1 Integrated USB-C Usage....155
25.2 Signal Description....156
25.3 Integrated Pull-Ups and Pull-Downs....156
25.4 I/O Signal Planes and States....157

26.0 Host System Management Bus (SMBus) Controller....158

26.1 Functional Description....158
26.1.1 Host Controller....158
26.1.2 SMBus Slave Interface....165
26.2 Signal Description....171
26.3 Integrated Pull-Ups and Pull-Downs....172
26.4 I/O Signal Planes and States....172
26.5 SMBus Power Gating....172

27.0 Serial Peripheral Interface (SPI)....173

27.1 Functional Description....173
27.1.1 SPI0 for Flash....173
27.1.2 SPI0 Support for TPM....177
27.2 Signal Description....178
27.3 Integrated Pull-Ups and Pull-Downs....178
27.4 I/O Signal Planes and States....179
27.5 VCCSPI Voltage (3.3 V or 1.8 V) Selection....179

28.0 Touch Host Controller (THC)....180

28.1 Functional Description....180
28.2 Signal Description....181
28.3 Integrated Pull-Ups and Pull-Downs....182
28.4 I/O Signal Planes and States....182

29.0 Intel® Serial IO Generic SPI (GSPI) Controllers....184

29.1 Functional Description....184
29.1.1 Controller Overview....184
29.1.2 DMA Controller....185
29.1.3 Reset....186
29.1.4 Power Management....186
29.1.5 Interrupts....187
29.1.6 Error Handling....187
29.2 Signal Description....187
29.3 Integrated Pull-Ups and Pull-Downs....188
29.4 I/O Signal Planes and States....188

30.0 Testability....189

30.1 Intel® Trace Hub (Intel® TH)....190
30.2 Intel® Direct Connect Interface (DCI)....190
30.2.1 Out Of Band (OOB) Hosting DCI....191
30.2.2 USB 3.2 Hosting DCI.DBC....191
30.2.3 Platform Setup....191
30.3 JTAG....191
30.3.1 Signal Description....192

30.3.2 I/O Signal Planes and States....193

31.0 Intel® Serial I/O Universal Asynchronous Receiver/Transmitter (UART) Controllers....194

31.1 Functional Description....195
31.1.1 UART Serial (RS-232) Protocols Overview....195
31.1.2 16550 8-bit Addressing - Debug Driver Compatibility....196
31.1.3 DMA Controller....196
31.1.4 Reset....197
31.1.5 Power Management....197
31.1.6 Interrupts....198
31.1.7 Error Handling....198
31.2 Signal Description....198
31.3 Integrated Pull-Ups and Pull-Downs....199
31.4 I/O Signal Planes and States....199
31.5 LSx....199
31.5.1 LSx Signal Description....199
31.5.2 Integrated Pull-Ups and Pull-Downs....200
31.5.3 I/O Signal Planes and States....200

32.0 Universal Serial Bus (USB)....201

32.1 Functional Description....201
32.1.1 eXtensible Host Controller Interface (xHCI) Controller....201
32.1.2 USB Dual Role Support - eXtensible Device Controller Interface (xDCI) Controller....201
32.1.3 AUX BIAS Control - USB Type-C Implementation with no Retimer....202
32.2 Signal Description....204
32.3 Integrated Pull-Ups and Pull-Downs....206
32.4 I/O Signal Planes and States....206
32.5 Supported USB 2.0 Ports....207

33.0 Connectivity Integrated (CNVi)....208

33.1 Functional Description....208
33.2 Signal Description....209
33.3 Integrated Pull-ups and Pull-downs....212
33.4 I/O Signal Planes and States....212

34.0 GPIO Serial Expander....213

34.1 Functional Description....213
34.2 Signal Description....214
34.3 Integrated Pull-ups and Pull-downs....214

35.0 Universal Flash Storage (UFS)....215

35.1 Functional Description....215
35.2 Signals Description....215
35.3 I/O Signals Planes and States....216

36.0 Private Configuration Space Target Port ID....217

37.0 Miscellaneous Signals....219

37.1 Signal Description....219
37.2 Integrated Pull-Ups and Pull-Downs....220
37.3 I/O Signal Planes and States....220

Figures

1 Flexible HSIO Lane Multiplexing in PCH-P 16
2 TCO Compatible Mode SMBus Configuration..... 29
3 Advanced TCO Mode..... 30
4 Basic eSPI Protocol..... 65
5 eSPI Slave Request to PCH for PCH Temperature..... 69
6 PCH Response to eSPI Slave with PCH Temperature 69
7 eSPI Slave Request to PCH for PCH RTC Time..... 70
8 PCH Response to eSPI Slave with RTC Time 71
9 Data Transfer on the I^2C Bus..... 80
10 Conceptual Diagram of SLP_LAN#..... 90
11 Integrated Clock Controller (ICC) Diagram..... 99
12 Generation of SERR# to Platform..... 106
13 PCH-P Supported PCI Express* Link Configurations 110
14 Conceptual Diagram of SLP_LAN#..... 132
15 Flow for Port Enable/Device Present Bits..... 150
16 Serial Data Transmitted over SGPIO Interface..... 153
17 Flash Descriptor Regions..... 175
18 VCCSPI Voltage (3.3 V or 1.8 V) Selection..... 179
19 THC Block Diagram..... 181
20 Platform Setup with Intel® Trace Hub 190
21 Platform Setup with DCI Connection..... 191
22 UART Serial Protocol 195
23 UART Receiver Serial Data Sample Points..... 196
24 GPIO - Virtual Wire Index Bit Mapping 203
25 Supported USB 2.0 Ports on the PCH..... 207
26 Example of GSX Topology..... 213

Tables

1 References..... 14
2 PCH I/O Capabilities 14
3 PCH SKUs.....14
4 PCH HSIO Details 15
5 PCH Device and Revision ID 17
6 PCH ACPI Device ID for GPIO Controller.....18
7 Fixed I/O Ranges Decoded by PCH.....20
8 Variable I/O Decode Ranges 22
9 PCH Memory Decode Ranges (Processor Perspective)..... 23
10 Boot Block Update Scheme..... 25
11 Acronyms.....27
12 Event Transitions that Cause Messages.....29
13 Legacy Replacement Routing.....33
14 Power Rail Descriptions for P PCH..... 39
15 Pin Straps.....40
16 Counter Operating Modes..... 45
17 Acronyms.....47
18 Signal Descriptions..... 51
19 Integrated Pull-Ups and Pull-Downs..... 54
20 I/O Signal Planes and States.....54
21 Acronyms.....60
22 Digital Display Signals.....60
23 Embedded DisplayPort* (eDP*) Backlight Control Signals.....61
24 Integrated Pull-Ups and Pull-Downs..... 62
25 I/O Signal Planes and States.....62
26 Acronyms.....64
27 References..... 64
28 eSPI Channels and Supported Transactions..... 67
29 eSPI Virtual Wires (VW).....68
30 Acronyms.....74
31 Native Function Signals Supporting Dynamic Termination Override..... 77
32 Acronyms.....79
33 References..... 79
34 Acronyms.....85
35 References..... 85
36 LAN Mode Support.....88
37 GbE LAN Signals.....90
38 Acronyms.....93
39 IPC Initiator -> Target flows.....95
40 Signal Descriptions..... 99
41 I/O Signal Pin States.....100
42 Acronym.....102
43 MSI Versus PCI IRQ Actions.....102
44 Power Plane and States for PCI Express* Signals 109
45 PCI Express* Port Feature Details 109
46 Acronyms..... 112
47 References.....112
48 General Power States for Systems Using the PCH.....113
49 State Transition Rules for the PCH 114
50 System Power Plane.....115
51 Causes of SMI and SCI 116
52 Sleep Types 119
53 Causes of Wake Events.....119
54 Transitions Due to Power Failure 121

55 Supported Deep Sx Policy Configurations 122
56 Deep Sx Wake Events 123
57 Transitions Due to Power Button..... 124
58 Write Only Registers with Read Paths in ALT Access Mode..... 128
59 PIC Reserved Bits Return Values..... 129
60 SUSPWRDNACK/SUSWARN#/GPP_A13 Pin Behavior..... 133
61 SUSPWRDNACK During Reset..... 134
62 Causes of Host and Global Resets..... 136
63 Acronyms..... 143
64 Acronyms..... 145
65 References..... 145
66 Acronyms..... 155
67 Acronyms..... 158
68 References..... 158
69 I^2C* Block Read..... 162
70 Enable for SMBALERT# 164
71 Enables for SMBus Slave Write and SMBus Host Events..... 164
72 Enables for the Host Notify Command..... 164
73 Slave Write Registers..... 166
74 Command Types..... 166
75 Slave Read Cycle Format..... 167
76 Data Values for Slave Read Registers..... 167
77 Host Notify Format..... 169
78 Data Values for Slave Read Registers..... 170
79 Enables for SMBus Slave Write and SMBus Host Events..... 171
80 Acronyms..... 173
81 SPI0 Flash Regions..... 174
82 Region Size Versus Erase Granularity of Flash Components 175
83 Region Access Control Table..... 176
84 Acronyms..... 180
85 Acronyms..... 184
86 Acronyms..... 189
87 References..... 190
88 Testability Signals..... 192
89 Power Planes and States for Testability Signals..... 193
90 Acronyms..... 195
91 Acronyms..... 201
92 References..... 201
93 Acronyms..... 208
94 References..... 208
95 Private Configuration Space Register Target Port IDs 217
96 Signal Descriptions..... 219
97 Integrated Pull-Ups and Pull-Downs..... 220

Revision History

Document Number	Revision Number	Description	Revision Date
763122	001	Initial Release	January 2023

1.0 Introduction

This document is intended for Original Equipment Manufacturers (OEMs), Original Design Manufacturers (ODM) and BIOS vendors creating products based on the Intel® 700 Series Chipset Family On-Package Platform Controller Hub (PCH).

NOTE

Throughout this document, the Platform Controller Hub (PCH) is used as a general term and refers to all Intel® 700 Series Chipset Family On-Package Platform Controller Hub (PCH) family I/O

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

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

1.1 Overview

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

- ACPI Power Management Logic Support, Revision 5.0a
- PCI Express Base Specification Revision 4.0
- Integrated Serial ATA Host controller 3.2, supports data transfer rates of up to 6 Gb/s on all ports
- USB 3.2 Gen 2x1 (10 Gb/s)eXtensible Host Controller (xHCI)
- USB 3.2 Gen 1x1 (5 Gb/s) Dual Role (eXtensible Device Controller - xDCI) Capability
- Serial Peripheral Interface (SPI)
- Enhanced Serial Peripheral Interface (eSPI)
- General Purpose Input Output (GPIO)
- Interrupt controller
- Timer functions
- System Management Bus (SMBus) Specification, Version 2.0
- Integrated Clock Controller (ICC)/Real Time Clock Controller (RTCC)
- Intel® High Definition Audio and Intel® Smart Sound Technology (Intel® SST), MIPI* SoundWire*, and DMIC.
- Intel® Serial I/O UART Host controllers

- Intel® Serial I/O I2C Host controllers
- Integrated 10/100/1000 Megabit Ethernet MAC
- Integrated Sensor Hub (ISH)
- Supports Intel® Rapid Storage Technology (Intel® RST)
- Supports Intel® Active Management Technology (Intel® AMT)
- Supports Intel® Virtualization Technology for Directed I/O (Intel® VT-d)
- Supports Intel® Trusted Execution Technology (Intel® TXT)
- JTAG Boundary Scan support
- Intel® Trace Hub (Intel® TH) and Direct Connect Interface (DCI) for debug
- Supports Intel® CSME
- Supports Integrated connectivity (CNVi)

Table 1. References

Specification	Document Number
Intel® 700 Series Chipset Family On-Package Platform Controller Hub (PCH) Datasheet, Volume 2 of 2	764889

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 2. PCH I/O Capabilities

Interface	P PCH
Processor Interface	OPI x8, up to 4 Gbps
Integrated GbE Controller	1 Data Link to Intel® Ethernet connection I219
eSPI	2 CS#, Quad Mode
Integrated Sensor Hub (ISH)	ISH 5.4

1.2 PCH SKUs

Table 3. PCH SKUs

Features	P PCH
PCIe Gen 3 Lanes	12
SATA 3.0 Ports (6 Gbps capable)	Up to 2
Integrated USB 2 Ports	10
Integrated USB 3.2 Ports	Up to 4
UFS2.1	1
Integrated Wi-Fi / Bluetooth*(CNVio)	Wi-Fi* 6 , Bluetooth*5.2
Display Audio	4 DSP Cores, 3 MB SRAM

Table 4. PCH HSIO Details

SKU	0	1	2	3	4	5	6	7	8	9	10	11
P PCH	PCIe , USB 3.2 Gen 1x1 /2 x1	PCIe , USB 3.2 Gen 1x1 /2 x1	PCIe , USB 3.2 Gen 1x1 /2 x1	PCIe , USB 3.2 Gen 1x1 /2 x1	PCIe	PCIe	PCIe; GbE	PCIe; GbE	PCIe; GbE; UFS	PCIe; UFS	PCIe; SATA	PCIe; SATA

1.3 Flexible High Speed I/O

Flexible Input/Output (I/O) is a technology that allows some of the PCH High Speed I/O (HSIO) lanes to be configured for connection to a Gigabit Ethernet (GbE) Controller, a PCIe* Controller, an Extensible Host Controller Interface (xHCI) USB 3.2 Controller, or an Advanced Host Controller Interface (AHCI) SATA Controller. Flexible I/O enables customers to optimize the allocation of the PCH HSIO interfaces to better meet the I/O needs of their system.

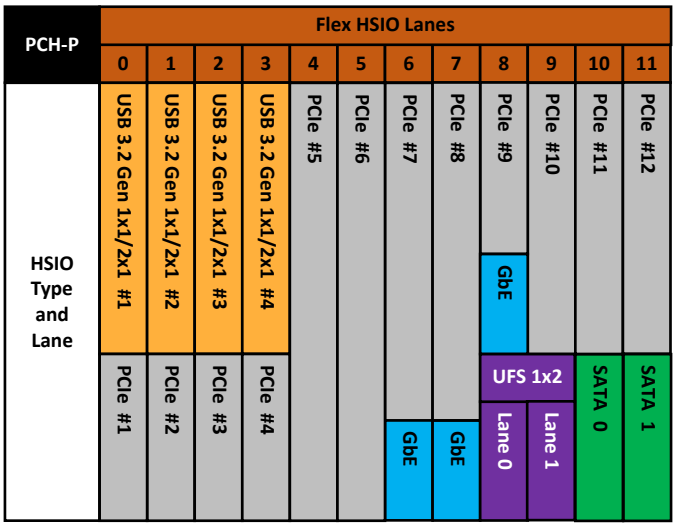
NOTE

Some Flexible I/O multiplexing capabilities are not available on all SKUs. Refer to the PCH SKUs on page 14 for specific SKU implementation details.

Acronyms	Description
USB	Universal Serial Bus
PCIe*	PCI Express* (Peripheral Component Interconnect Express*)
GbE	Gigabit Ethernet
SATA	Serial Advanced Technology Attachment
HSIO	High Speed Input/Output

1.3.1 PCH-P

Figure 1. Flexible HSIO Lane Multiplexing in PCH-P



The 12 Flexible HSIO Lanes [11:0] on PCH-P support the following configurations:

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

       --> Max PCIe* Root Ports (or devices) = 6 - GbE (0 or 1)
   - PCIe* Lanes 1-4 (PCIe* Controller #1), 5-8 (PCIe* Controller #2), and 9-12 (PCIe* Controller #3) must be individually configured.
2. Up to four USB 3.2 Gen 1x1/2x1 Lanes
   - A maximum of four USB 3.2 Gen 1x1/2x1 Ports (or devices) can be enabled.
   - USB 3.2 Gen 1x1 = 5 GT/s
   - USB 3.2 Gen 2x1 = 10 GT/s
3. Up to two SATA Lanes
   - A maximum of two SATA Ports (or devices) can be enabled.
4. Up to three GbE Lanes
   - A maximum of one GbE Port (or device) can be enabled.
5. Supports two Lane 1x2 Universal Flash Storage (UFS)
6. For unused USB 3.2/PCIe* Combo Lanes, the unused lanes must be statically assigned to PCIe* or USB 3.2 via the USB 3.2/PCIe* Combo Port Soft Straps discussed in the SPI Programming Guide and through the Intel Flash Image Tool (FIT) tool.

2.0 PCH Controller Device IDs

2.1 Device and Revision ID

The Revision ID (RID) register is an 8-bit register located at offset 08h in the PCI header of every PCI/PCIe* function.

Table 5. PCH Device and Revision ID

P Dev ID	Device Function - Device Description	Note
5180 - 519F	D31:F0 - eSPI Controller	**PCH Device ID :** P : 519D
51A0	D31:F1 - P2SB	
51A1	D31:F2 - PMC	
51A3	D31:F4 - SMBus	
51A4	D31:F5 - SPI (flash) Controller	
15FB	D31:F6 - GbE Controller: Corporate/Intel® vPro™ (Default)	
15FC	D31:F6 - GbE Controller: Consumer	
51A6	D31:F7 - Intel® Trace Hub (Intel® TH)	
51A8	D30:F0 - UART #0	
51A9	D30:F1 - UART #1	
51AA	D30:F2 - GSPI #0	
51AB	D30:F3 - GSPI #1	
51B0	D29:F0 - PCI Express* Root Port #9	
51B1	D29:F1 - PCI Express* Root Port #10	
51B2	D29:F2 - PCI Express* Root Port #11	
51B3	D29:F3 - PCI Express* Root Port #12	
51B8	D28:F0 - PCI Express* Root Port #1	
51B9	D28:F1 - PCI Express* Root Port #2	
51BA	D28:F2 - PCI Express* Root Port #3	
51BB	D28:F3 - PCI Express* Root Port #4	
51BC	D28:F4 - PCI Express* Root Port #5	
51BD	D28:F5 - PCI Express* Root Port #6	
51BE	D28:F6 - PCI Express* Root Port #7	
51BF	D28:F7 - PCI Express* Root Port #8	

continued...

P Dev ID	Device Function - Device Description	Note
51C5	D25:F0 - I2C Controller #4	
51C6	D25:F1 - I2C Controller #5	
51C7	D25:F2 - UART #2	
51C8 - 51CF	D31:F3 - Intel® High Definition Audio (Intel®HD Audio) (Audio, Voice, Speech)	HDA Host Device ID 51CA
51D0	D16:F6 - Touch Host Controller #0 (THC #0)	
51D1	D16:F7 - Touch Host Controller #1 (THC #1)	
51D3	D23:F0 - SATA Controller (AHCI)	
51D7	D23:F0 - SATA Controller (RAID 0/1/5/10) - premium	
282A	D23:F0 - SATA Controller (RAID 0/1/5/10) - In- box Compatible ID	
51D8	D16:F0 - I2C Controller #6	
51D9	D16:F1 - I2C Controller #7	
51DA	D17:F0 - UART Controller #3	
51E0	D22:F0 - Intel®CSME: HECI #1	
51E1	D22:F1 - Intel® CSME: HECI #2	
51E2	D22:F2 - Intel® CSME: IDE Redirection (IDE-R)	
51E3	D22:F3 - Intel® CSME: Keyboard and Text (KT) Redirection	
51E4	D22:F4 - Intel® CSME: HECI #3	
51E5	D22:F5 - Intel® CSME: HECI #4	
51E8	D21:F0 - I2C Controller #0	
51E9	D21:F1 - I2C Controller #1	
51EA	D21:F2 - I2C Controller #2	
51EB	D21:F3 - I2C Controller #3	
51ED	D20:F0 - USB 3.2 Gen 2x1 (10 Gb/s) xHCI HC	
51EE	D20:F1 - USB 3.2 Gen 1x1 (5 Gb/s) Device Controller (xDCI)	
51EF	D20:F2 - Shared SRAM	
51F0 - 51F3	D20:F3 - CNVi: Wi-Fi*	
51FB	D18:F6 - GSPI #2	
51FC	D18:F0 - Integrated Sensor Hub	
51FF	D18:F0 - UFS	

Table 6. PCH ACPI Device ID for GPIO Controller

ACPI ID
P : INTC1055

3.0 Memory Mapping

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

3.1 Functional Description

This section provides information on the following topics:

- PCI Devices and Functions
- Fixed I/O Address Ranges
- Variable I/O Decode Ranges

3.1.1 PCI Devices and Functions

The PCH incorporates a variety of PCI devices and functions, as shown in the following table. If for some reason, the particular system platform does not want to support any one of the Device Functions, with the exception of D30:F0, they can individually be disabled. The integrated Gigabit Ethernet controller will be disabled if no Platform LAN Connect component is detected (Gigabit Ethernet Controller on page 85). When a function is disabled, it does not appear to the software. A disabled function will not respond to any register reads or writes, ensuring that these devices appear hidden to software.

NOTE

The reference to DMI for the PCH SKU is On Package DMI (OPI).

3.1.2 Fixed I/O Address Ranges

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

NOTE

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

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

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

Table 7. Fixed I/O Ranges Decoded by PCH

I/O Address	Read Target	Write Target	Internal Unit (unless[E]: External)[2]	Separate Enable/Disable
20h – 21h	Interrupt Controller	Interrupt Controller	Interrupt	None
24h – 25h	Interrupt Controller	Interrupt Controller	Interrupt	None
28h – 29h	Interrupt Controller	Interrupt Controller	Interrupt	None
2Ch – 2Dh	Interrupt Controller	Interrupt Controller	Interrupt	None
2E-2F	Super I/O	Super I/O	[E] Forwarded to eSPI	Yes. ESPI_IOD_IOE.SE
30h – 31h	Interrupt Controller	Interrupt Controller	Interrupt	None
34h – 35h	Interrupt Controller	Interrupt Controller	Interrupt	None
38h – 39h	Interrupt Controller	Interrupt Controller	Interrupt	None
3Ch – 3Dh	Interrupt Controller	Interrupt Controller	Interrupt	None
40h	Timer/Counter	Timer/Counter	8254 Timer	None
42h-43h	Timer/Counter	Timer/Counter	8254 Timer	None
4E-4F	Microcontroller	Microcontroller	[E] Forwarded to eSPI	Yes. ESPI_IOD_IOE.ME2
50h	Timer/Counter	Timer/Counter	8254 Timer	None
52h-53h	Timer/Counter	Timer/Counter	8254 Timer	None
60h	Keyboard Controller	Keyboard Controller	[E] Forwarded to eSPI	Yes, with 64h. ESPI_IOD_IOE.KE
61h	NMI Controller	NMI Controller	CPU I/F	None
62h	Microcontroller	Microcontroller	[E] Forwarded to eSPI	Yes, with 66h. ESPI_IOD_IOE.ME1
63h	NMI Controller [1]	NMI Controller [1]	CPU I/F	Yes, alias to 61h. GIC.P61AE
64h	Keyboard Controller	Keyboard Controller	[E] Forwarded to eSPI	Yes, with 60h. ESPI_IOD_IOE.KE
65h	NMI Controller [1]	NMI Controller [1]	CPU I/F	Yes, alias to 61h. GIC.P61AE
66h	Microcontroller	Microcontroller	[E] Forwarded to eSPI	Yes, with 62h. ESPI_IOD_IOE.ME1
67h	NMI Controller [1]	NMI Controller [1]	CPU I/F	Yes, alias to 61h. GIC.P61AE
70h	RTC Controller	NMI and RTC Controller	RTC	None
71h	RTC Controller	RTC Controller	RTC	None
72h	RTC Controller	RTC Controller	RTC	None. Alias to 70h if RC.UE[4]=0, else 72h
73h	RTC Controller	RTC Controller	RTC	None. Alias to 71h if

continued...

I/O Address	Read Target	Write Target	Internal Unit (unless[E]: External)[2]	Separate Enable/Disable
				RC.UE='0', else 73h
74h	RTC Controller	RTC Controller	RTC	None
75h	RTC Controller	RTC Controller	RTC	None
76h-77h	RTC Controller	RTC Controller	RTC	None. Alias to 70h-71h if RC.UE=0, else 76h-77h
80h[3]	eSPI or PCIe	eSPI or PCIe	Read: [E] eSPI or PCIe Write: [E] eSPI or [E] PCIe	None. PCIe if GCS.RPR='1', else eSPI
84h - 86h	eSPI or PCIe	eSPI or PCIe	Read: [E] eSPI or PCIe Write: [E] eSPI or [E] PCIe	None. PCIe if GCS.RPR='1', else eSPI
88h	eSPI or PCIe	eSPI or PCIe	Read: [E] eSPI or PCIe Write: [E] eSPI or [E] PCIe	None. PCIe if GCS.RPR='1', else eSPI
8Ch - 8Eh	eSPI or PCIe	eSPI or PCIe	Read: [E] eSPI or PCIe Write: [E] eSPI or [E] PCIe	None. PCIe if GCS.RPR='1', else eSPI
90h	eSPI	eSPI	Read: [E] eSPI Write: [E] eSPI	None. Alias to 80h
92h	Reset Generator	Reset Generator	CPU I/F	None
94h - 96h	eSPI	eSPI	Read: [E] eSPI Write: [E] eSPI	None. Alias to 8xh
98h	eSPI	eSPI	Read: [E] eSPI Write: [E] eSPI	None. Alias to 88h
9Ch - 9Eh	eSPI	eSPI	Read: [E] eSPI Write: [E] eSPI	None. Alias to 8xh
A0h - A1h	Interrupt Controller	Interrupt Controller	Interrupt	None
A4h - A5h	Interrupt Controller	Interrupt Controller	Interrupt	None
A8h - A9h	Interrupt Controller	Interrupt Controller	Interrupt	None

continued...

I/O Address	Read Target	Write Target	Internal Unit (unless[E]: External)[2]	Separate Enable/Disable
ACh - ADh	Interrupt Controller	Interrupt Controller	Interrupt	None
B0h - B1h	Interrupt Controller	Interrupt Controller	Interrupt	None
B2h - B3h	Power Management	Power Management	Power Management	None
B4h - B5h	Interrupt Controller	Interrupt Controller	Interrupt	None
B8h - B9h	Interrupt Controller	Interrupt Controller	Interrupt	None
BCh - BDh	Interrupt Controller	Interrupt Controller	Interrupt	None
200-207h	Gameport Low	Gameport Low	Forwarded to eSPI	Yes. ESPI_CS1IORE.LGE
208-20Fh	Gameport High	Gameport High	Forwarded to eSPI	Yes. ESPI_CS1IORE.HGRE
4D0h – 4D1h	Interrupt Controller	Interrupt Controller	Interrupt Controller	None
CF9h	Reset Generator	Reset Generator	Interrupt controller	None

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

3.1.3 Variable I/O Decode Ranges

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

WARNING

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

Table 8. Variable I/O Decode Ranges

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

continued...

Range Name[1]	Mappable	Size (Bytes)	Target
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
Note: 1. All ranges are decoded directly from			

3.2 Memory Map

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

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

Table 9. PCH Memory Decode Ranges (Processor Perspective)

Memory Range	Target	Dependency/Comments
000E 0000 - 000E FFFF	eSPI or SPI	Bit 6 in BIOS Decode Enable Register is set
000F 0000 - 000F FFFF	eSPI or SPI	Bit 7 in BIOS Decode Enable Register is set
FECX X000 - FECX X040	I/O(x)APIC inside PCH	XX controlled via APIC Range Select (ASEL) field and APIC Enable (AEN) bit
FEC1 0000 - FEC1 7FFF	PCIe port 1	PCIe root port 1 I/OxApic Enable (PAE) is set
FEC1 8000 - FEC1 FFFF	PCIe port 2	PCIe root port 2 I/OxApic Enable (PAE) is set
FEC2 0000 - FEC2 7FFF	PCIe port 3	PCIe root port 3 I/OxApic Enable (PAE) is set
FEC2 8000 - FEC2 FFFF	PCIe port 4	PCIe root port 4 I/OxApic Enable (PAE) is set
FEC3 0000 - FEC3 7FFF	PCIe port 5	PCIe root port 5 I/OxApic Enable (PAE) is set
FEC3 8000 - FEC3 FFFF	PCIe port 6	PCIe root port 6 I/OxApic Enable (PAE) is set
FEC4 0000 - FEC4 7FFF	PCIe port 7	PCIe root port 7 I/OxApic Enable (PAE) is set
FEC4 8000 - FEC4 FFFF	PCIe port 8	PCIe root port 8 I/OxApic Enable (PAE) is set
FEC5 0000 - FEC5 7FFF	PCIe port 9	PCIe root port 9 I/OxApic Enable (PAE) is set
FEC5 8000 - FEC5 FFFF	PCIe port 10	PCIe root port 10 I/OxApic Enable (PAE) is set
FEC6 0000 - FEC6 7FFF	PCIe port 11	PCIe root port 11 I/OxApic Enable (PAE) is set
FEC6 8000 - FEC6 FFFF	PCIe port 12	PCIe root port 12 I/OxApic Enable (PAE) is set
FEF0 0000 - FEFF FFFF	eSPI or SPI	uCode Patch Region Enable UCPR.UPRE is set
FFC0 0000 - FFC7 FFFF FF80 0000 - FF87 FFFF	eSPI or SPI	Bit 8 in BIOS Decode Enable Register is set
FFC8 0000 – FFCF FFFF FF88 0000 - FF8F FFFF	eSPI or SPI	Bit 9 in BIOS Decode Enable Register is set

continued...

Memory Range	Target	Dependency/Comments
FFD0 0000 - FFD7 FFFF FF90 0000 - FF97 FFFF	eSPI or SPI	Bit 10 in BIOS Decode Enable Register is set
FFD8 0000 – FFDF FFFF FF98 0000 - FF9F FFFF	eSPI or SPI	Bit 11 in BIOS Decode Enable Register is set
FFE0 0000 - FFE7 FFFF FFA0 0000 - FFA7 FFFF	eSPI or SPI	Bit 12 in BIOS Decode Enable Register is set
FFE8 0000 – FFEF FFFF FFA8 0000 – FFAF FFFF	eSPI or SPI	Bit 13 in BIOS Decode Enable Register is set
FFF0 0000 - FFF7 FFFF FFB0 0000 - FFB7 FFFF	eSPI or SPI	Bit 14 in BIOS Decode Enable Register is set
FFFC 0000 - FFFF FFFF	eSPI, SPI, or Intel® CSME	Always enabled. Refer to Table 10 on page 25 for swappable ranges
FFF8 0000 - FFFB FFFF FFB8 0000 - FFBF FFFF	eSPI or SPI	Always enabled. Refer to Table 10 on page 25 for swappable ranges
FF70 0000 - FF7F FFFF FF30 0000 - FF3F FFFF	eSPI or SPI	Bit 3 in BIOS Decode Enable Register is set
FF60 0000 - FF6F FFFF FF20 0000 - FF2F FFFF	eSPI or SPI	Bit 2 in BIOS Decode Enable Register is set
FF50 0000 - FF5F FFFF FF10 0000 - FF1F FFFF	eSPI or SPI	Bit 1 in BIOS Decode Enable Register is set
FF40 0000 - FF4F FFFF FF00 0000 - FF0F FFFF	eSPI or SPI	Bit 0 in BIOS Decode Enable Register is set
FED0 X000 - FED0 X3FF	HPET	BIOS determines “fixed” location which is one of four 1 KB ranges where X (in the first column) is 0h, 1h, 2h, or 3h
FED4 0000 - FED4 7FFF	SPI (set by strap)	TPM and Trusted Mobile KBC
FED4 C000 - FED4 FFFF	PCH Internal (PSF Error Handler)	Always enabled
FED6 0000 – FED6 1FFF	PCH Internal (Intel® Trace Hub (Intel® TH)/xHCI)	Always enabled
FED6 2000 – FED6 3FFF	xHCI (Processor)	Fixed range in CPU – never forwarded to PCH
FED5 0000 - FED5 FFFF	Intel® CSME	Always enabled
FED7 0000 - FED7 4FFF	Internal Device	Security feature related
128 KB anywhere in 4 GB range	LAN Controller (CSR registers)	Enable via standard PCI mechanism (Device 31:Function 6)
4 KB anywhere in 4 GB range	LAN Controller (LAN space on Flash)	Enable via standard PCI mechanism (Device 31:Function 6)
64 KB anywhere in 64-bit address range	USB Host Controller	Enable via standard PCI mechanism (Device 20, Function 0)
2 MB anywhere in 4 GB range	USB Device Controller	Enable via standard PCI mechanism (Device 20, Function 1)
24 KB anywhere in 4 GB range	USB Device Controller	Enable via standard PCI mechanism (Device 20, Function 1)
16 KB anywhere in 64-bit addressing space	Intel® HD Audio Subsystem	Enable via standard PCI mechanism (Device 31, Function 3)

continued...

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)
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
16 Bytes anywhere in 64-bit address range	Intel® CSMEI #1, #2, #3, #4	Enable via standard PCI mechanism
4 KB anywhere in 4 GB range	Intel® AMT Keyboard and Text	Enable via standard PCI mechanism (Device 22: Function 3)
16 MB anywhere in 64-bit address range	P2SB	Enable via standard PCI mechanism
Eight 4 KB slots anywhere in 64-bit address range	UART, GPI and I2C controllers	Enable via standard PCI mechanism
1 MB (BAR0) or 4 KB (BAR1) in 4GB range	Integrated Sensor Hub	Enable via standard PCI mechanism (Device 19: Function 0)
8 KB slot anywhere in 4 GB range	Integrated Wi-Fi*	Enable via standard PCI mechanism
8 KB slot and 4 KB slot anywhere in 4 GB range	PMC	Enable via standard PCI mechanism
8 KB slot and 4 KB slot anywhere in 4 GB range	Shared SRAM	Enable via standard PCI mechanism

3.2.1 Boot Block Update Scheme

The PCH supports a "Top-Block Swap" mode that has the PCH swap the top block in the 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 appropriate address lines as selected in Boot Block Size (BOOT_BLOCK_SIZE) soft strap for SPI.

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 10. Boot Block Update Scheme

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

continued...

BOOT_BLOCK_SIZE Value	Accesses to	Being Directed to
100 (1 MB)	FFF0_0000h - FFFF_FFFFh	FFE0_0000h - FFEF_FFFFh and vice versa
101 (2 MB)	FFE0_0000h - FFFF_FFFFh	FFC0_0000h - FFDF_FFFFh and vice versa
110 (4 MB)	FFC0_0000h - FFFF_FFFFh	FF80_0000h - FFBF_FFFFh and vice versa
111 (8 MB)	FF80_0000h - FFFF_FFFFh	FF00_0000h - FF7F_FFFFh and vice versa
Note: This bit is automatically set to 0 by RTCRST#, but not by PLTRST#.		

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

The algorithm is:

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

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

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

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

The BIOS algorithm should be as follows:

1. If an RTC well power failure is experienced during a boot block update, the system will probably not be able to boot at that point.
2. The user can set the Top-Swap pin strap and force the system to boot from the 2nd block. The code in the 2nd block should read the valid BIOS image from disk and put it into the top-swap.
3. The BIOS will not be able to clear the Top-Swap bit (because the jumper is in place). The user should then remove the jumper and reboot.

4.0 System Management

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

The following features and functions are supported by the PCH:

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

Table 11. Acronyms

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

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.

4.1.1 Handling an Intruder

The PCH has an input signal, INTRUDER#, that can be attached to a switch that is activated when the system's case is 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

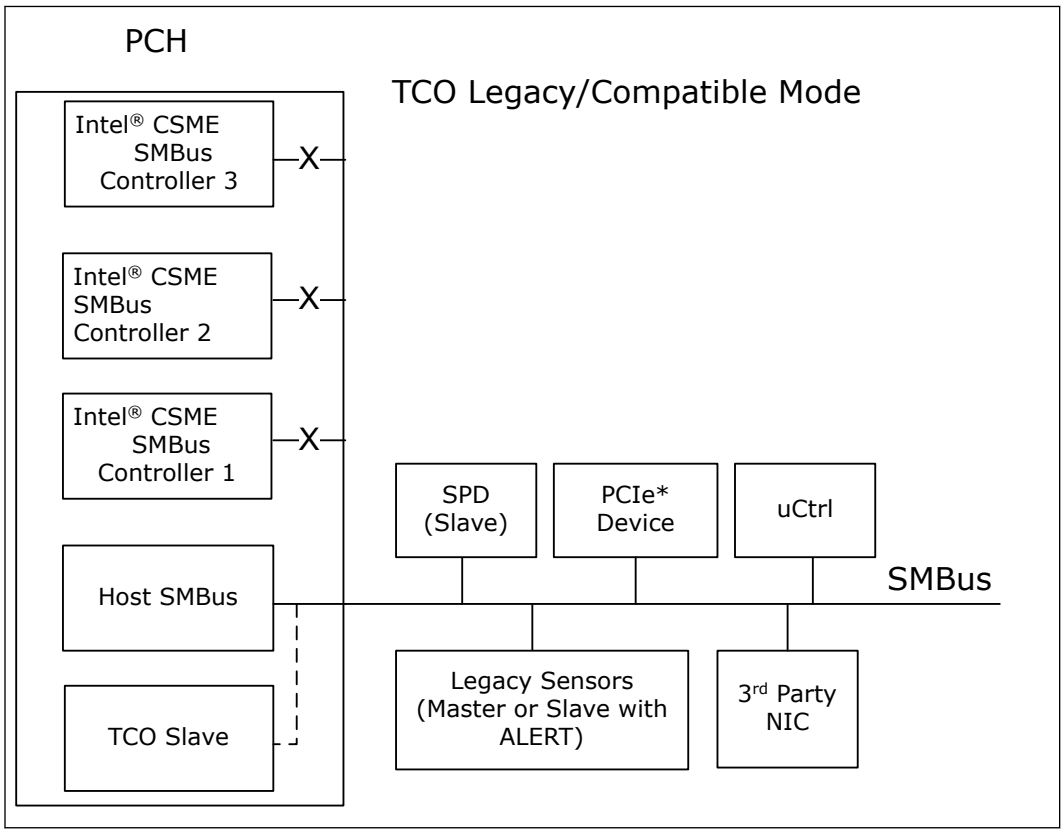
This is slightly different than a classic sticky bit, since most sticky bits would remain active indefinitely when the signal goes active and would immediately go inactive when a 1 is written to the bit.

4.1.2 TCO Modes

TCO Compatible Mode

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

Figure 2. 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. The table below includes a list of events that will report messages to the network management console.

Table 12. Event Transitions that Cause Messages

Event	Assertion?	Deassertion?	Comments
INTRUDER# pin	Yes	No	System must hung in S0 state
Watchdog Timer Expired	Yes	NA	System will enter to hung state
SMBALERT# pin	Yes	Yes	System must hung in S0 state
BATLOW#	Yes	Yes	System must hung in S0 state
CPU_PWR_FLR	Yes	No	System will enter to hung state

Advanced TCO Mode

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

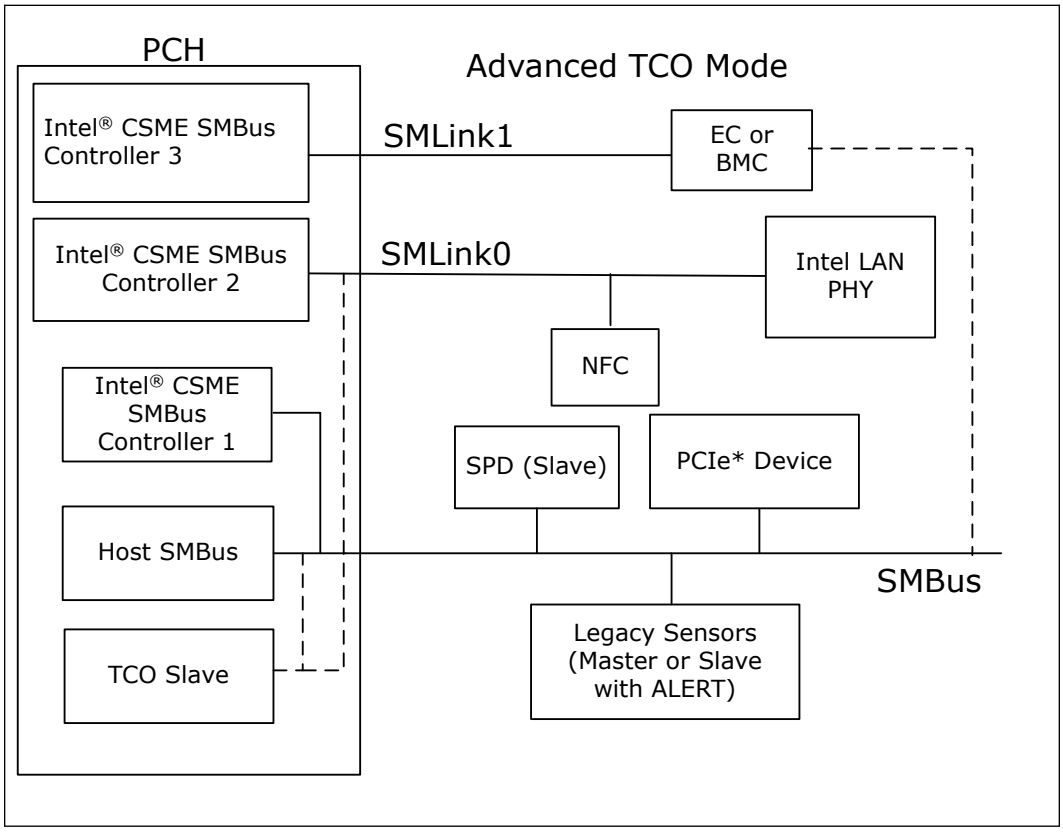
In this mode, the Intel® CSME SMBus controllers must be enabled by soft strap in the flash descriptor. Refer to the figure below for more details.

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

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

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

Figure 3. Advanced TCO Mode



5.0 High Precision Event Timer (HPET)

5.1 Feature Overview

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

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

Timer 0 supports periodic interrupts.

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

5.1.1 Timer Accuracy

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

Within any 100 us period, the timer reports a time that is up to two ticks too early or too late. Each tick is less than or equal to 100 ns; thus, this represents an error of less than 0.2%.

The timer is monotonic. It does not return the same value on two consecutive reads (unless the counter has rolled over and reached the same value).

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

5.1.2 Timer Off-load

The PCH supports a timer off-load feature that allows the HPET timers to remain operational during very low power S0 operational modes when the PCH's XTAL clock is disabled. The clock source during this off-load is the Real Time Clock's 32.768 kHz clock. This clock is calibrated against the PCH's XTAL clock during boot time to an accuracy that ensures the error introduced by this off-load is less than 10 ppb (0.000001%).

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

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

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

Theory of Operation

The Off-loadable Timer Block consists of a 64 bit fast clock counter and an 82 bit 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 us 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 64 bits of the 82 bit counter, with the 18 LSBs set to zero. The actual transition though 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 64 bits of an 82 bit 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 64 bits of the slow clock value as the fast count value. It then de-asserts the indication that slow clock mode is active. The 32 kHz clock counter no longer counts. The 64 bit MSB will be over-written when the 32 kHz counter is reloaded once conditions are met to enable the 32 kHz HPET counter but the 18 bit LSB is retained and it is not cleared out during the next reload cycle to avoid losing the fractional part of the counter.

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

5.1.3 Interrupt Mapping

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

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 below table.

Table 13. 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.			

Mapping Option #2 (Standard Option)

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

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

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

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

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

NOTE

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

NOTE

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

Mapping Option #3 (Processor Message Option)

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

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

NOTE

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

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

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

5.1.4 Periodic Versus Non-Periodic Modes

Non-Periodic Mode

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

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

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

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

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

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 us.

All of the timers support non-periodic mode.

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

Periodic Mode

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

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

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

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

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

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

Software sets the ENABLE_CNF bit to enable interrupts.

NOTE

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

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

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

5.1.5 Enabling the Timers

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

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

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

5.1.6 Interrupt Levels

Interrupts directed to the internal 8259s are active high. Refer to the **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.

For handling interrupts and issues related to 64 bit timers with 32 bit processors, refer to IA-PC HPET Specification.

6.0 PCH Thermal Sensor

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

6.1 Modes of Operation

The DTS has two usages when enabled:

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

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

6.2 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.

6.3 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.
- No accuracy is specified for temperature range beyond 110 °C or below -10 °C.

6.4 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 SMBus and/or over eSPI. If enabled, PMC will drive the temperature directly to the SMBus and eSPI units . 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 127°C, where 255 (0xFF) indicates that the sensor isn't enabled yet. The EC must be connected to either SMLink1 or eSPI for thermal reporting support.

NOTE

The catastrophic trip value is set close to 110 °C and is not programmed or accessible by BIOS.

6.5 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 PHLC register) is compared to the present temperature. If the present temperature is greater than the PHLC 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.

7.0 Power Delivery

This section provides information on the following topics:

- Power and Ground Signals

7.1 Power and Ground Signals

This section describes the PCH power rails.

Table 14. Power Rail Descriptions for P PCH

Name	Description
VCCIN_AUX	FIVR Input rail: 1.8 V
VCC_VNNEXT_1P05	Used for FIVR PRIM_CORE bypass mode during S0ix and Sx: 1.05 V
VCC_V1P05EXT_1P05	Used for FIVR PCH IO bypass mode during S0ix and Sx: 1.05 V
VCCA_CLKLDO_1P8	Analog supply for internal clocks: 1.8 V
VCCPRIM1P05_OUT_PCH	1.05 V Primary Well: for CNVi and other internal I/O blocks.
VCCDSW_1P05	Deep Sx Well: 1.05 V. This rail is generated by on die DSW low dropout (LDO) linear regulator to supply DSW core logic.
VCCPRIM_1P8	1.8 V Primary Well. Note: When the VCCPRIM_1P8 is off and the VCCPRIM_3P3 is powered on during G3 to S5, there may be a leakage current from the VCCPRIM_3P3 power rail to VCCPRIM_1P8 power rail.
VCCPRIM_3P3	3.3 V Primary Well.
VCCPGPPR	Audio Power 3.3 V or 1.8 V. If powered at 3.3 V, the 3.3 V supply can come from VCCPRIM_3P3 supply. If powered at 1.8 V, the 1.8 V supply can come from VCCPRIM_1P8 supply.
VCCDSW_3P3	3.3 V Deep Sx Well.
VCCRTC	RTC Well Supply. This rail can drop to 2.0 V if all other planes are off. This power is not expected to be shut off unless the RTC battery is removed or drained. *Notes:* 1. VCCRTC nominal voltage is 3.0 V. 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. 2. Implementation should not attempt to clear CMOS by using a jumper to pull VCCRTC low. Clearing CMOS can be done by using a jumper on RTCRST# or GPI.
VCCDPHY_1P24	1.24 V for CNVi logic. This rail is generated internally with a LDO and needs to be routed to the motherboard so that the rail can be supplied back to the SoC.
VCCLDOSTD_0P85	This rail is generated internally and needs to be routed out to the motherboard for decoupling purpose.
VCC1P05_OUT_FET	FIVR output rail: 1.05 V, used for CPU rails VCCST/STG.
VSSINAUX_SENSE	VCCIN_AUX VSS sense pin.
VCCINAUX_SENSE	VCCIN_AUX sense pin.
VSS	Ground

8.0 Pin Straps

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

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

Table 15. Pin Straps

Signal	Usage	When Sampled	Comment
GPP_B14 / SPKR	Top Swap Override	Rising edge of PCH_PWROK	The strap has a 20 kohm ± 30% internal pull-down. 0=>Disable "Top Swap" mode. (Default) 1=>Enable "Top Swap" mode. This inverts an address on access to SPI, so the alternate boot block is fetched instead of the original boot-block. The PCH will invert the appropriate address lines (A[23:16]) as selected in Top Swap Block size soft strap. *Notes:* 1. The internal pull-down is disabled after PCH_PWROK is high. 2. Software will not be able to clear the Top Swap bit until the system is rebooted. 3. The status of this strap is readable using the Top Swap bit (Bus0, Device31, Function0, offset DCh, bit4). 4. This signal is in the primary well.
GPP_I18 / GSPI0_MOSI	No Reboot	Rising edge of PCH_PWROK	The strap has a 20 kohm ± 30% internal pull-down. 0=>Disable "No Reboot" mode. (Default) 1=>Enable "No Reboot" mode (PCH will disable the TCO Timer system reboot feature). This function is useful when running ITP/XDP. *Notes:* 1. The internal pull-down is disabled after PCH_PWROK is high. 2. This signal is in the primary well.
GPP_C2 / SMBALERT#	TLS Confidentiality	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0=>Disable Intel® CSME Crypto Transport Layer Security (TLS) cipher suite (no confidentiality). (Default) 1=>Enable Intel® CSME Crypto Transport Layer Security (TLS) cipher suite (with confidentiality). Must be pulled up to support Intel® AMT with TLS. *Notes:* 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
GPP_C5 / SML0ALERT#	eSPI Disable	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0 = Enable eSPI. (Default)

continued...

Signal	Usage	When Sampled	Comment
			1 = Disable eSPI. *Notes:* 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
SPI0_MOSI	Reserved	Rising edge of RSMRST#	External pull-up is required. Recommend 4.7 kohm pull-up. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.
GPP_B23 / SML1ALERT# / PCHHOT#	XTAL Frequency Selection Bit 1	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. This is bit 1 (MSB) of the 2-bit encoded pin straps for XTAL Frequency Selection. This strap is used in conjunction with XTAL Frequency Selection Bit 0 on GPP_J2 / CNV_BRI_DT / UART0_RTS# pin. 2-bit XTAL Frequency Selection encodings: 00 = 24 MHz (default) 01 = Reserved 10 = 38.4 MHz 11 = 25 MHz *Notes:* 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
SPI0_IO2	Reserved	Rising edge of RSMRST#	External pull-up is required. Recommend 100 kohm if pulled up to 3.3 V or 75 kohm if pulled up to 1.8 V. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.
SPI0_IO3	Reserved	Rising edge of RSMRST#	External pull-up is required. Recommend 100 kohm if pulled up to 3.3 V or 75 kohm if pulled up to 1.8 V. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.
GPP_R2 / HDA_SDO / HDACPU_SDO	Reserved	Rising edge of PCH_PWROK	This strap has a 20 kohm ± 30% internal pull-down. *Notes:* 1. The internal pull-down is disabled after PCH_PWROK is high. 2. This signal is in the primary well.
GPP_H12 / SML2ALERT#	eSPI Flash Sharing Mode	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0=>Master Attached Flash Sharing (MAFS) enabled (Default) 1=>Slave Attached Flash Sharing (SAFS) enabled. *Notes:* 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
GPP_H15 / SML3ALERT#	JTAG ODT Disable	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0=> JTAG ODT is disabled (default) 1=> JTAG ODT is enabled *Notes:* 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
GPP_H18 / SML4ALERT#	VCCSPI Voltage Configuration	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0=> VCCSPI at 3.3 V (Default) 1=> VCCSPI at 1.8 V *Notes:* 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.

continued...

Signal	Usage	When Sampled	Comment
DBG_PMODE	Reserved	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-up. This strap should sample high. There should NOT be any on-board device driving it to opposite direction during strap sampling. *Notes:* 1. The internal pull-up is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
GPD7	Reserved	Rising edge of DSW_PWROK	This strap has a 20 kohm ± 30% internal pull-down. This strap should sample LOW. There should NOT be any on-board device driving it to opposite direction during strap sampling. *Notes:* 1. The internal pull-down is disabled after DSW_PWROK is high. 2. This signal is in the DSW well.
GPP_J2 / CNV_BRI_DT / UART0_RTS#	XTAL Frequency Selection Bit 0	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. This is bit 0 (LSB) of the 2-bit encoded pin straps XTAL Frequency Selection . This strap is used in conjunction with XTAL Frequency Selection Bit 1 on GPP_B23 / SML1ALERT# / PCHHOT# pin. 2-bit XTAL Frequency Selection encodings: 00 = 24 MHz (default) 01 = Reserved 10 = 38.4 MHz 11 = 25 MHz *Notes:* 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
GPP_J4 / CNV_RGI_DT / UART0_TXD	M.2 CNVi Mode Select	Rising edge of RSMRST#	This strap does not have an internal pull-up or pull-down. A weak external pull-up is required. 0=>Integrated CNVi enabled. 1=>Integrated CNVi disabled. *Note:* When a RF companion chip is connected to the PCH CNVi interface, the device internal pull-down resistor will pull the strap low to enable CNVi interface.
GPP_I22 / GSPI1_MOSI	Boot BIOS Strap (BBS)	Rising edge of PCH_PWROK	This signal has a 20 kohm ± 30% internal pull-down. 0=>BIOS fetches are routed to SPI (MAF) or the eSPI Flash Channel (SAF) 1=>BIOS fetches are routed to the eSPI Peripheral Channel *Notes:* 1. The internal pull-down is disabled after PCH_PWROK de-asserts. 2. This signal is in the primary well.

9.0 Electrical and Thermal Characteristics

For more information, download the pdf, click on the navigation pane and refer the spreadsheet, **763122_001_Electr_Therm_Spec.xlsx**.

10.0 8254 Timers

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

Counter 0, System Timer

This counter functions as the system timer by controlling the state of IRQ0 and is typically programmed for Mode 3 operation. The counter produces a square wave with a period equal to the product of the counter period (838 ns) and the initial count value. The counter loads the initial count value 1 counter period after software writes the count value to the counter I/O address. The counter initially asserts IRQ0 and decrements the count value by two each counter period. The counter negates IRQ0 when the count value reaches 0. It then reloads the initial count value and again decrements the initial count value by two each counter period. The counter then asserts IRQ0 when the count value reaches 0, reloads the initial count value, and repeats the cycle, alternately asserting and negating IRQ0.

Counter 2, Speaker Tone

This counter provides the speaker tone and is typically programmed for Mode 3 operation. The counter provides a speaker frequency equal to the counter clock frequency (1.193 MHz) divided by the initial count value. The speaker must be enabled by a write to port 061h (Refer to the NMI Status and Control ports).

10.1 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 2-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.

The table below lists the six operating modes for the interval counters:

Table 16. 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.

10.2 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 one 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 2-byte counts, 2-bytes must be read. The2-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.

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.

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 2-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.

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 the 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 the 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.

11.0 Audio Voice and Speech

Table 17. Acronyms

Acronyms	Description
DMA	Direct Memory Access.
DMIC	Digital Microphone. PDM based MEMs microphone modules.
DSP	Digital Signal Processor. In AVS specifically a DSP to process audio data.
MEMs	Micro electrical mechanical Systems. For AVS devices such as Digital MEMs Microphones.
MSI	Message Signaled Interrupt. An in-band method of signaling an interrupt.
PCM	Pulse Code Modulation. Modulation with amplitude coded into stream.
PDM	Pulse Density Modulation. Modulation with amplitude coded by pulse density.
SDI	Serial Data In.
SDO	Serial Data Out.
SoC	System On Chip.
VOIP	Voice Over Internet Protocol

11.1 Feature Overview

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

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

The key HW features of the AVS Subsystem are described in the following topics:

- Intel® High Definition Audio (Intel® HD Audio) Controller Capabilities
- Audio DSP Capabilities
- Intel® High Definition Audio Interface Capabilities
- Direct Attached Digital Microphone (PDM) Interface
- USB Audio Offload Support
- Intel® Display Audio Interface
- MIPI* SoundWire* Interface

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

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

- Baseline Intel® HD Audio operation with legacy DMA transporting audio stream to / from audio codecs, with host CPU carrying out the audio processing
- Low power audio operation with offload DMA transporting audio stream to / from the Audio DSP offload engine offloading the audio processing from host CPU
- Ability transport audio stream to various audio codecs speaking different link protocols with the same audio host controller view from SW stacks
- PCI / PCI Express* controller
- Supports data transfers, descriptor fetches, and DMA position writes using VC0
- Independent Bus Master logic for 16 general purpose DMA streams: 7 input and 9 output
- Supports variable length stream slots
- Each audio stream supports up to:
  - 30 streams (15 input, 15 output)
  - 7 input DMA streams and 9 output DMA streams
  - 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)
- Support Converged Platform Power Management (CPPM)
  - Support 1 ms of buffering with all DMA running with maximum bandwidth.
  - Support 10 ms of buffering with 1 output DMA and 1 input DMA running at 2 channels, 96 kHz, 16 bit audio.

11.1.2 Audio DSP Capabilities

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

- Audio DSP based on 4Cadence* Tensilica* LX6 HIFI3 DSP Cores operating up to 400 MHz with 3 MB SRAM
- Low power support for Intel® Wake on Voice (Intel® WOV)
- Low power audio playback with post processing
- Low power VOIP and circuit switch voice call with pre-processing
- Low power FM radio playback
- Various DSP functions provided by DSP Core: MP3, AAC, 3rd Party IP Algorithms, etc.
- Host downloadable DSP FW functions
- Voice call processing enhancement
- Sensors algorithm offload / assistance: motion, ambient light, fingerprint, proximity, etc.
- Communication hub offload / assistance: location, peer device detection, Wi-Fi availability, etc.

11.1.3 Intel® High Definition Audio Interface Capabilities

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

- Two SDI signals to support two external codecs
- Drives variable frequency (6 MHz to 24 MHz) BCLK to support:
  - SDO double pumped up to 48 Mb/s
  - SDIs single pumped up to 24 Mb/s
- Provides cadence for 44.1 kHz-based sample rate output
- Supports 1.8 V and 3.3 V I/O voltages
  - 1.8 V and 3.3 V drive strengths has separate programming.

11.1.4 Direct Attached Digital Microphone (PDM) Interface

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

- Two DMIC PDM interfaces with each interface capable of supporting up to 2 digital MEMs microphones.
- Low power always listening support for Intel® Wake on Voice
- 2 PCM audio streams (with independent PCM sampling rate: 48 kHz or 16 kHz) per digital mic interface

- Ultrasound reception capable with higher frequency ranges between 3.84 MHz - 4.8 MHz.
- Support of 1.8 V I/O voltages

11.1.5 USB Audio Offload Support

USB Audio Offload provides audio mixing / processing support for USB audio endpoint connected through the xHCI Controller. This is aimed at providing a universal audio offload power benefit across various audio devices connected to the platform and USB audio usage is expected to gain more popularity with the introduction of USB Type-C* connector. These USB audio endpoint will be enumerated by the xHCI Controller SW and only the audio streaming path is peer to the Audio DSP subsystem for DSP FW mixing / processing support. USB Audio Offload capabilities are listed as follows:

- Up to 2 audio output streams support
- Up to 4 audio input streams support
- Provides cadence for 44.1 kHz-based sample rate output
- Support isochronous audio stream offload for LS / FS / HS USB audio device
- Support synchronous / asynchronous / adaptive modes of isochronous audio streaming
- Support non-PCM encoded audio bit stream defined by IEC61937 / IEC60958 standard
  - Packetizing into PCM sample format and PCM equivalent rates
- Single audio playback (synchronous / adaptive) at 4 ch x 192 KHz x 24 bits
- Support isochronous audio stream offload for LS / FS / HS USB audio device
- Single audio playback (asynchronous) at 8 ch x 48 KHz x 24 bits + single audio sync input at 1 ch x 1 KHz x 32 bits
- Up to 2 concurrent audio playback (synchronous / adaptive) at 8 ch x 96 KHz x 24 bit + 4ch x 48 KHz x 24 bit
- Single audio capture (synchronous / asynchronous) at 4 ch x 96 KHz x 24 bits
- Up to 2 concurrent audio capture (synchronous / asynchronous) of 8 ch x 48 KHz x 24 bit + audio sync input at 4 ch x 48 KHz x 24 bit

11.1.6 Intel® Display Audio Interface

The Intel® Display Audio link do not offer external pin out to the package, but internally between PCH and Processor and offers connection to the Intel® Display Audio codec that usually resides in the Processor. The Intel® Display Audio codec provides audio stream routing to the integrated HDMI and DP links through the existing Intel® HD Audio controller SW stacks. The Intel® Display Audio codec is enumerated by the Intel® HD Audio driver software if discovered over the Intel Display Audio Interface.

11.1.7 MIPI® SoundWire* Interface

The SoundWire interface is an optional feature offering connection to the SoundWire devices, which include audio codecs and modem codecs. The SoundWire interface is the latest audio interface targeting (but not limited to) the phone and tablet market and the main advantage is the connection simplicity with a two wires multi-drop

topology + PCM/PDM streaming capabilities. Currently SoundWire devices are non-standard across different vendors , hence it is very likely to require customized audio codec SW per vendor. These devices will be enumerated based on vendor / device ID of the SoundWire device reporting. SoundWire interface capabilities are listed as follows:

- 4 independent SoundWire Interfaces with multi-drop connections to audio peripherals
- Single audio playback at 8 ch x 96 kHz x 24 bits
- Up to 2 concurrent audio playback at 2 ch x 192 kHz x 24 bit each
- Single audio capture at 8 ch x 96 kHz x 24 bits
- Up to 4 concurrent audio capture of 2 ch x 96 kHz x 24 bit each
- Up to 4 x SoundWire interfaces frame rate synchronized on global periodic events
- Up to 6 x PCM bidirectional streams per SoundWire interface
  - — Direction is programmable as either input or output stream
- 4 x PDM input streams per SoundWire interface
- Up to 2 channels per PCM streams
- Up to 1 channel per PDM streams
- Ability to map each stereo PCM streams to a sub-set of a multi-channel PCM stream DMA data transferred over Audio Link Hub
- Ability to map each mono PDM input stream to a sub-set of a multi-channel PDM stream DMA data transferred to digital mic port (decimation input)
- Supports 1.8 V I/O voltages

11.2 Signal Description

Table 18. Signal Descriptions

Signal Name	Type	Description
Intel High Definition Audio Signals		
GPP_R4 / **HDA_RST#** / I2S2_SCLK / DMIC_CLK_A0	O	**Intel HD Audio Reset**: Master H/W reset to internal/ external codecs.
GPP_R1 / **HDA_SYNC** / I2S0_SFRM / DMIC_CLK_B1	O	**Intel HD Audio Sync**: 48 kHz fixed rate frame sync to the codecs. Also used to encode the stream number.
GPP_R0 / **HDA_BCLK** / I2S0_SCLK / DMIC_CLK_B0 / HDA_PROC_BCLK	O	**Intel HD Audio Bit Clock**: Up to 24 MHz serial data clock generated by the Intel HD Audio controller.
GPP_R2 / **HDA_SDO** / I2S0_TXD / HDA_PROC_SDO	O	**Intel HD Audio Serial Data Out**: Serial TDM data output to the codecs. The serial output is double-pumped for a bit rate of up to 48 Mb/s.
GPP_R3 / **HDA_SDI0** / I2S0_RXD / HDA_PROC_SDI	I	**Intel HD Audio Serial Data In 0**: Serial TDM data input from the two codec(s). The serial input is single-pumped for a bit rate of up to 24 Mb/s. These signals contain integrated Pull-down resistors, which are enabled while the primary well is powered.

continued...

Signal Name	Type	Description
GPP_R5 / **HDA_SDI1** / I2S2_SFRM / DMIC_DATA0	I	**Intel HD Audio Serial Data In 1**: Serial TDM data input from the two codec(s). The serial input is single-pumped for a bit rate of up to 24 Mb/s. These signals contain integrated Pull-down resistors, which are enabled while the primary well is powered.
Intel Display Audio Interface		
GPP_R0 / HDA_BCLK / I2S0_SCLK / DMIC_CLKB0 / **HDA_PROC_BCLK**	O	**Display Audio Bit Clock**: Serial data clock generated by the Intel HD Audio controller. PCH supports data rate of up to 96 Mb/s.
GPP_R2 / HDA_SDO / I2S0_TXD / **HDA_PROC_SDO**	O	**Display Audio Serial Data Out**: Serial TDM data output to the codec. PCH supports data rate of up to 96 Mb/s.
GPP_R3 / HDA_SDI0 / I2S0_RXD / **HDA_PROC_SDI**	I	**Display Audio Serial Data In**: Serial TDM data input from the codec. PCH supports data rate of up to 96 Mb/s.
I^2S/PCM Interface		
GPP_R0 / HDA_BCLK / **I2S0_SCLK** / DMIC_CLKB0 / HDA_PROC_BCLK	I/O	**I^2S/PCM serial bit clock 0**:Clock used to control the timing of a transfer. Can be generated internally (Master mode) or taken from an external source (Slave mode).
GPP_S0 / SNDW0_CLK / **I2S1_SCLK**	I/O	**I^2S/PCM serial bit clock 1**:This clock is used to control the timing of a transfer. Can be generated internally (Master mode) or taken from an external source (Slave mode).
GPP_R4 / HDA_RST# / **I2S2_SCLK** / DMIC_CLKA0	I/O	**I^2S/PCM serial bit clock 2**:This clock is used to control the timing of a transfer. Can be generated internally (Master mode) or taken from an external source (Slave mode).
GPP_R1 / HDA_SYNC / **I2S0_SFRM** / DMIC_CLKB1	I/O	**I^2S/PCM serial frame indicator 0**: This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Master mode) or taken from an external source (Slave mode).
GPP_S1 / SNDW0_DATA / **I2S1_SFRM**	I/O	**I^2S/PCM serial frame indicator 1**: This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Master mode) or taken from an external source (Slave mode).
GPP_R5 / HDA_SDI1 / **I2S2_SFRM** / DMIC_DATA0	I/O	**I^2S/PCM serial frame indicator 2**: This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Master mode) or taken from an external source (Slave mode).
GPP_R2 / HDA_SDO / **I2S0_TXD** / HDA_PROC_SDO	O	**I^2S/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.
GPP_S2 / SNDW1_CLK / DMIC_CLKA0 / **I2S1_TXD**	O	**I^2S/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.
GPP_R6 / **I2S2_TXD** / DMIC_CLKA1	O	**I^2S/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.
GPP_R3 / HDA_SDI0 / **I2S0_RXD** / HDA_PROC_SDI	I	**I^2S/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.

continued...

Signal Name	Type	Description
GPP_S3 / SNDW1_DATA / DMIC_DATA0 / **I2S1_RXD**	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.
GPP_R7 / **I2S2_RXD** / DMIC_DATA1	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.
GPP_D19 / **I2S_MCLK1_OUT**	O	**I²S/PCM Master reference clock 1**: This signal is the master reference clock that connects to an audio codec.
DMIC Interface		
GPP_S2 / SNDW1_CLK / **DMIC_CLKA0** / I2S1_TXD or GPP_R4 / HDA_RST# / I2S2_SCLK / **DMIC_CLKA0**	O	**Digital Mic Clock A0**:Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz. May be duplicated into CLKA and CLKB for individual left / right DMIC power control.
GPP_S6 / SNDW3_CLK / **DMIC_CLKA1** or GPP_R6 / I2S2_TXD / **DMIC_CLKA1**	O	**Digital Mic Clock A**1:Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz. May be duplicated into CLKA and CLKB for individual left / right DMIC power control.
GPP_S4 / SNDW2_CLK / **DMIC_CLKB0** or GPP_R0 / HDA_BCLK / I2S0_SCLK / **DMIC_CLKB0** / HDA_PROC_BCLK	O	**Digital Mic Clock B0**:Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz. May be duplicated into CLKA and CLKB for individual left / right DMIC power control.
GPP_S5 / SNDW2_DATA / DMIC_CLKB1 or GPP_R1 / HDA_SYNC / I2S0_SFRM / **DMIC_CLKB1**	O	**Digital Mic Clock B**1:Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz. May be duplicated into CLKA and CLKB for individual left / right DMIC power control.
GPP_S3 / SNDW1_DATA / **DMIC_DATA0** / I2S1_RXD or GPP_R5 / HDA_SDI1 / I2S2_SFRM / **DMIC_DATA0**	I	**Digital Mic Data**:Serial data input from the digital mic.
GPP_S7 / SNDW3_DATA / **DMIC_DATA1** or GPP_R7 / I2S2_RXD / **DMIC_DATA1**	I	**Digital Mic Data**:Serial data input from the digital mic.
SoundWire Interface		
GPP_S0 / **SNDW0_CLK** / I2S1_SCLK	I/O	**SoundWire Clock**: Serial data clock to external peripheral devices.
GPP_S1 / **SNDW0_DATA** / I2S1_SFRM	I/O	**SoundWire Data**: Serial data input from external peripheral devices.
GPP_S2 / **SNDW1_CLK** / DMIC_CLKA0 / I2S1_TXD	I/O	**SoundWire Clock**: Serial data clock to external peripheral devices.
GPP_S3 / **SNDW1_DATA** / DMIC_DATA0 / I2S1_RXD	I/O	**SoundWire Data**: Serial data input from external peripheral devices.
GPP_S4 / **SNDW2_CLK** / DMIC_CLKB0	I/O	**SoundWire Clock**: Serial data clock to external peripheral devices.

continued...

Signal Name	Type	Description
GPP_S5 / **SNDW2_DATA** / DMIC_CLKB1	I/O	**SoundWire Data**: Serial data input from external peripheral devices.
GPP_S6 / **SNDW3_CLK** / DMIC_CLKA1	I/O	**SoundWire Clock**: Serial data clock to external peripheral devices.
GPP_S7 / **SNDW3_DATA** / DMIC_DATA1	I/O	**SoundWire Data**: Serial data input from external peripheral devices.
SNDW_RCOMP	I/O	**SoundWire RCOMP**:200ohm +/- 1% compensation resistor required to ground.
Misc		
GPP_B14 / **SPKR** / TIME_SYNC1 / SATA_LED# / ISH_GP6	O	**Speaker Output**:Used for connection to external speaker for POST sounds if not using HD_Audio embedded option.

11.3 Integrated Pull-Ups and Pull-Downs

Table 19. Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value (Ω)
HDA_SYNC	Pull-down	20 kohm
HDA_SDO	Pull-down	20 kohm
HDA_SDI[1:0]	Pull-down	20 kohm
HDACPU_SDO	Pull-down	20 kohm
HDACPU_SDI	Pull-down	20 kohm
I2S[2:0]_SCLK	Pull-down	20 kohm
I2S[2:0]_SFRM	Pull-down	20 kohm
I2S[2:0]_TXD	Pull-down	20 kohm
I2S[2:0]_RXD	Pull-down	20 kohm
I2S_MCLK1_OUT	Pull-down	20 kohm
DMIC_DATA[1:0]	Pull-down	20 kohm
SNDW[3:0]_DATA	Pull-down	5 kohm
SPKR	Pull-down	20 kohm

11.4 I/O Signal Planes and States

Table 20. I/O Signal Planes and States

Signal Name	Power Plane	During Reset[2]	Immediately After Reset[2]	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_BCLK	Primary	Driven Low	Driven Low	Driven Low	OFF

continued...

Signal Name	Power Plane	During Reset[2]	Immediately After Reset[2]	S4/S5	Deep Sx
HDA_SDO	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
HDA_SDI[1:0]	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I²S/PCM Interface					
I2S[2:0]_SCLK	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S[2:0]_SFRM	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S[2:0]_TXD	Primary	Driven Low	Driven Low	Driven Low	OFF
I2S[2:0]_RXD	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S_MCLK1_OUT	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
DMIC Interface					
DMIC_CLKA[1:0]	Primary	Driven Low	Driven Low	Driven Low	OFF
DMIC_CLKB[1:0]	Primary	Driven Low	Driven Low	Driven Low	OFF
DMIC_DATA[1:0]	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SoundWire Interface					
SNDW[3:0]_DATA	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SNDW[3:0]_CLK	Primary	Driven Low	Driven Low	Driven Low	OFF
Misc					
SPKR	Primary	Internal Pull-down	Driven Low	Low then disabled (refer to note)	OFF

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

12.0 Controller Link

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

Acronyms	Description
CL	Controller Link
WLAN	Wireless Local Area Network

12.1 Signal Description

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

12.2 Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value (Ohm)	Notes
CL_DATA	Pull-up Pull-down	31.25 100	I/O Signal Planes and States on page 56
CL_CLK	Pull-up Pull-down	31.25 100	

12.3 I/O Signal Planes and States

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

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

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

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

1. Driven mode: To drive "1" on this pin, Controller Link turn-on the output enable and output=1 to drive 3.3 V on this pin. This mode can only be enabled with older version of WLAN which is 3.3 V tolerant.
2. Open-drain mode: To drive "1", Controller Link turn-off the output-enable, and external (required) pull-up will pull the pin up to 1.8 V, which is compatible with WLAN voltage requirement.

13.0 Processor Sideband Signals

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

Acronyms	Description
PECI	Platform Environmental Control Interface

13.1 Functional Description

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

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

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

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

13.2 Signal Description

Signal 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
GPP_E3 / **PROC_GP0**	I	Thermal management signal
GPP_E7 / **PROC_GP1**	I	Thermal management signal
GPP_B3 / **PROC_GP2** / ISH_GP4B	I	Thermal management signal
GPP_B4 / **PROC_GP3** / ISH_GP5B	I	Thermal management signal

13.3 Integrated Pull-Ups and Pull-Downs

None

13.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset [1]	S4/S5	Deep Sx
PROCPWRGD	Primary	Undriven	Driven High	OFF	OFF
THERMTRIP#	Primary	Undriven	Undriven	OFF	OFF
PECI	Primary	Undriven	Undriven	OFF	OFF
PROC_GP[3:0]	Primary	Undriven	Undriven	Undriven	OFF
Note: 1. Reset reference for primary well pins is RSMRST#.					

14.0 Digital Display Signals

Table 21. Acronyms

Acronyms	Description
eDP*	embedded Display Port*

14.1 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. Seven pairs of DDC (DDP*_CTRLCLK and DDP*_CTRLDATA) signals exist on the PCH that correspond to seven 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*. DDC and HDP signals are muxed with GPIO pads that can be independently configured to 1.8 V or 3.3 V via soft straps. Therefore, depending on soft strap settings for the corresponding GPIO pads, DDC and HDP signals can support either 1.8 V or 3.3 V.

Table 22. Digital Display Signals

Name	Type	Description
GPP_E14/**DDSP_HPDA**/DISP_MISCA	I	**Display Port A** : HPD Hot-Plug Detect.
GPP_A18/**DDSP_HPDB**/DISP_MISCB	I	**Display Port B** : HPD Hot-Plug Detect.
GPP_A19/**DDSP_HPD1**/DISP_MISC1	I	**Display Port 1** : HPD Hot-Plug Detect.
GPP_A20/**DDSP_HPD2**/DISP_MISC2	I	**Display Port 2** : HPD Hot-Plug Detect.
GPP_A14/USB_OC1#/**DDSP_HPD3**/DISP_MISC3	I	**Display Port 3** : HPD Hot-Plug Detect.
GPP_A15/USB_OC2#/**DDSP_HPD4**/DISP_MISC4	I	**Display Port 4** : HPD Hot-Plug Detect.

continued...

Name	Type	Description
GPP_E22/**DDPA_CTRLCLK**/DNX_FORCE_RELOAD	I	**Display Port A** : Control Clock.
GPP_E23/**DDPA_CTRLDATA**	O	**Display Port A** : Control Data.
GPP_H15/**DDPB_CTRLCLK**/PCIE_LINK_DOWN	I	**Display Port B** : Control Clock.
GPP_H17/**DDPB_CTRLDATA**	O	**Display Port B** : Control Data.
GPP_A21/**DDPC_CTRLCLK**	I	**Display Port C** : Control Clock.
GPP_A22/**DDPC_CTRLDATA**	O	**Display Port C** : Control Data.
GPP_E18/**DDP1_CTRLCLK**/TBT_LSX0_TXD/ BSSB_LS0_RX	I	**Display Port 1** : Control Clock.
GPP_E19/**DDP1_CTRLDATA**/TBT_LSX0_RXD/ BSSB_LS0_TX	O	**Display Port 1** : Control Data.
GPP_E20/**DDP2_CTRLCLK**/TBT_LSX1_TXD/ BSSB_LS1_RX	I	**Display Port 2** : Control Clock.
GPP_E21/**DDP2_CTRLDATA**/TBT_LSX1_RXD/ BSSB_LS1_TX	O	**Display Port 2** : Control Data.
GPP_D9/ISH_SPI_CS#/**DDP3_CTRLCLK**/ TBT_LSX2_TXD/BSSB_LS2_RX/GSPI2_CS0#	I	**Display Port 3** : Control Clock.
GPP_D10/ISH_SPI_CLK/**DDP3_CTRLDATA**/ TBT_LSX2_RXD/BSSB_LS2_TX/GSPI2_CLK	O	**Display Port 3** : Control Data.
GPP_D11/ISH_SPI_MISO/**DDP4_CTRLCLK**/ TBT_LSX3_TXD/BSSB_LS3_RX/GSPI2_MISO	I	**Display Port 4** : Control Clock.
GPP_D12/ISH_SPI_MOSI/**DDP4_CTRLDATA**/ TBT_LSX3_RXD/BSSB_LS3_TX/GSPI2_MOSI	O	**Display Port 4** : Control Data.

14.2 Embedded DisplayPort* (eDP*) Backlight Control Signals

Table 23. Embedded DisplayPort* (eDP*) Backlight Control Signals

Signal Name	Type	Description
VDDEN	O	**Primary eDP Panel power Enable** : Panel power control enable. This signal is used to control the VDC source of the panel logic.
eDP_BKLTCTL	O	**Primary eDP Backlight Enable** : Panel backlight enable control for eDP This signal is used to gate power into the backlight circuitry.
eDP_BKLTEN	O	**Primary eDP Panel Backlight Brightness control** : Panel brightness control for eDP. This signal `is used as the PWM Clock input signal.
GPP_A17/**DISP_MISCC**	O	**Secondary eDP Panel power Enable** : Panel power control enable. This signal is used to control the VDC source of the panel logic.
GPP_A21/**DDPC_CTRLCLK**	O	**Secondary eDP Backlight Enable** : Panel backlight enable control for eDP This signal is used to gate power into the backlight circuitry.
GPP_A22/**DDPC_CTRLDATA**	O	**Secondary eDP Panel Backlight Brightness control** : Panel brightness control for eDP. This signal `is used as the PWM Clock input signal.
Note: VDDEN, eDP_BKLTEN, eDP_BKLTCTL, DISP_MISCC, DDPC_CTRLCLK, DDPC_CTRLDATA can be left as no connect if eDP* is not used.		

14.3 Integrated Pull-Ups and Pull-Downs

Table 24. Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value	Notes
DDPA_CTRLDATA	Pull-down	15-40 kohm	Refer to the note below
DDPB_CTRLDATA	Pull-down	15-40 kohm	
DDPC_CTRLDATA	Pull-down	15-40 kohm	
DDP1_CTRLDATA	Pull-down	15-40 kohm	
DDP2_CTRLDATA	Pull-down	15-40 kohm	
DDP3_CTRLDATA	Pull-down	15-40 kohm	
DDP4_CTRLDATA	Pull-down	15-40 kohm	
Note: The internal pull-up/pull-down is only applied during the strap sampling window (PCH_PWROK) and is then disabled. Enabling can be done using a 2.2 kohm Pull-up resistor.			

14.4 I/O Signal Planes and States

Table 25. I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
DDSP_HPDA	Primary	Undriven	Undriven	Undriven	OFF
DDSP_HPDB	Primary	Undriven	Undriven	Undriven	OFF
DDSP_HPD1	Primary	Undriven	Undriven	Undriven	OFF
DDSP_HPD2	Primary	Undriven	Undriven	Undriven	OFF
DDSP_HPD3	Primary	Undriven	Undriven	Undriven	OFF
DDSP_HPD4	Primary	Undriven	Undriven	Undriven	OFF
DDPA_CTRLCLK	Primary	Undriven	Undriven	Undriven	OFF
DDPA_CTRLDATA	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
DDPB_CTRLCLK	Primary	Undriven	Undriven	Undriven	OFF
DDPB_CTRLDATA	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
DDP1_CTRLCLK	Primary	Undriven	Undriven	Undriven	OFF
DDP1_CTRLDATA	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
DDP2_CTRLCLK	Primary	Undriven	Undriven	Undriven	OFF
DDP2_CTRLDATA	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
DDP3_CTRLCLK	Primary	Undriven	Undriven	Undriven	OFF
DDP3_CTRLDATA	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
DDP4_CTRLCLK	Primary	Undriven	Undriven	Undriven	OFF

continued...

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
DDP4_CTRLDATA	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
eDP_VDDEN	Primary	Driven Low	Driven Low	Driven Low	OFF
eDP_BKLTEN	Primary	Driven Low	Driven Low	Driven Low	OFF
eDP_BKLTCTL	Primary	Driven Low	Driven Low	Driven Low	OFF
DISP_MISCC	Primary	Driven Low	Driven Low	Driven Low	OFF
DDPC_CTRLCLK	Primary	Driven Low	Driven Low	Driven Low	OFF
DDPC_CTRLDATA	Primary	Driven Low	Driven Low	Driven Low	OFF
Note: 1. Reset reference for primary well pins is RSMRST#.					

15.0 Enhanced Serial Peripheral Interface (eSPI)

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

- 1.8 V support only
- Support for Master Attached Flash and Slave Attached Flash.
- Support for up to 50 MHz (configured by soft straps)
- Up to quad mode support
- Support for PECI over eSPI
- Support for Multiple OOB Master (dedicated OOB channel for different OOB masters in the PCH such as PMC and Intel® CSME)
- Transmitting RTC time/date to the slave device upon request
- In-band messages for communication between the PCH and slave device to eliminate side-band signals.
- Real time SPI flash sharing, allowing real time operational access by the PCH and slave device.

Table 26. Acronyms

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

Table 27. References

Specification	Document Number/Location
Enhanced Serial Peripheral Interface (eSPI) Specifications	https://downloadcenter.intel.com/download/27055/eSPI

15.1 Functional Description

This section provides information on the following topics:

- Operating Frequency
- Protocols
- WAIT States from eSPI Slave
- In-Band Link Reset
- Slave Discovery
- Flash Sharing Mode

- PECI Over eSPI
- Multiple OOB Master
- Channels and Supported Transactions

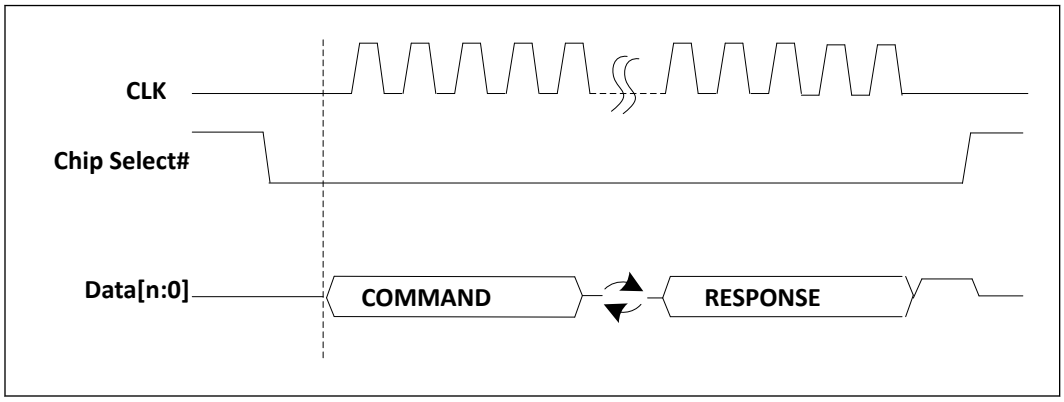
15.1.1 Operating Frequency

The eSPI controller supports 20 MHz, 25 MHz, 33 MHz, and 50 MHz. A slave device can support frequencies lower than the recommended maximum frequency (50 MHz). In addition, the slave device must support a minimum frequency of 20 MHz for default (reset) communication between the Master and Slave device.

15.1.2 Protocols

Below is an overview of the basic eSPI protocol. Refer to the latest eSPI Specification and corresponding platform eSPI Compatibility Specification for more details (Refer to Table 27 on page 64).

Figure 4. Basic eSPI Protocol



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

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

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

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

15.1.3 WAIT States from eSPI Slave

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

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

15.1.4 In-Band Link Reset

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

15.1.5 Slave Discovery

The PCH eSPI interface is enabled using a hard pin strap. Refer to Pin Straps on page 40 for details on the strap.

If eSPI interface is disabled via Hardware strap , the eSPI controller will gate all its clocks and put itself to sleep.

15.1.6 Flash Sharing Mode

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

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

15.1.7 PECI Over eSPI

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

The PECI bus may be connected to the PCH via either the legacy PECI pin or the eSPI interface. Either of the operation via legacy PECI pin or over eSPI is enabled at a time in a given platform.

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

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

15.1.8 Multiple OOB Master

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

15.1.9 Channels and Supported Transactions

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

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

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

Table below summarizes the eSPI channels and supported transactions.

Table 28. 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

Peripheral Channel (Channel 0) Overview

The Peripheral channel performs the following functions:

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

Virtual Wire Channel (Channel 1) Overview

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

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

  Table below summarizes the PCH virtual wires in eSPI mode.

Table 29. 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
SUSWARN#	Output	ESPI_RESET#	No
SUS_ACK	Input	ESPI_RESET#	No
SUSPWRDNACK	Output	ESPI_RESET#	No
PLTRST#	Output	ESPI_RESET#	Yes
PME# (eSPI Peripheral PME)	Input	ESPI_RESET#	N/A
WAKE#	Input	ESPI_RESET#	No
SMI#	Input	PLTRST#	N/A
SCI#	Input	PLTRST#	N/A
RCIN#	Input	PLTRST#	No
SLP_A#	Output	ESPI_RESET#	Yes
SLP_S4#/SLP_S5#/ SLP_LAN#/SLP_WLAN#	Output	DSW_PWROK	Yes
SLAVE_BOOT_LOAD_DONE	Input	ESPI_RESET#	N/A
SLAVE_BOOT_LOAD_STATUS	Input	ESPI_RESET#	N/A
HOST_RST_WARN	Output	PLTRST#	N/A
HOST_RST_ACK	Input	PLTRST#	N/A
OOB_RST_WARN	Output	ESPI_RESET#	N/A
OOB_RST_ACK	Input	ESPI_RESET#	N/A
HOST_C10	Output	PLTRST#	N/A
ERROR_NONFATAL	Input	ESPI_RESET#	N/A
ERROR_FATAL	Input	ESPI_RESET#	N/A

- **Interrupt Events**

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

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

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

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

Out-of-Band Channel (Channel 2) Overview

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

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

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

  The packet formats for the temperature request from the eSPI slave and the PCH response back are shown in the two figures below.

Figure 5. eSPI Slave 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. = 01h (PCH OOB HW Handler)							0
4	Common code = 01h (Get_PCH_Temp)							
5	Byte Count = 01h							
6	Source Slave Address[7:0] = 0Fh (eSPI Slave 0/EC)							1

Figure 6. PCH Response to eSPI Slave 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. = 0Eh (eSPI Slave 0/EC)							0
4	Common code = 01h (Get_PCH_Temp)							
5	Byte Count = 02h							
6	Source Slave Address [7:0] = 01h (PCH OOB HW Handler)							1
7	PCH Temperature Data [7:0]							

- **PCH RTC Time/Date to EC Over eSPI OOB Channel**

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

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

The packet formats for the RTC time/date request from the eSPI slave and the PCH response back to the device are shown in the two figures below.

Figure 7. eSPI Slave 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	Dest Slave Addr. [7:1] = 01h (PCH OOB HW Handler)							0
4	Common code = 02h (Get_PCH_RTC_Time)							
5	Byte Count = 01h							
6	Source Slave Address [7:0] = 0Fh (eSPI Slave 0/EC)							1

Figure 8. PCH Response to eSPI Slave 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	Dest Slave Addr. [7:0] = 0Eh (eSPI Slave 0/EC)							0
4	Common code = 02h (Get_PCH_RTC_Time)							
5	Byte Count = 09h							
6	Source Slave Address [7:1] = 01h (PCH OOB HW Handler)							1
7	Reserved					DM	HF	DS
8	PCH RTC Time: Seconds							
9	PCH RTC Time: Minutes							
10	PCH RTC Time: Hours							
11	PCH RTC Time: Day of Week							
12	PCH RTC Time: Day of Month							
13	PCH RTC Time: Month							
14	PCH 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.

Flash Access Channel (Channel 3) Overview

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

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

- **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.

- **Slave Attached Flash Channel Controller (SAFCC) Flash Operation and Addressing**

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

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

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

15.2 Signal Description

Signal Name	Type	Description
GPP_A0 / **ESPI_IO0**	I/O	**eSPI Data Signal 0:** Bi-directional pin used to transfer data between the PCH and eSPI device.
GPP_A1 / **ESPI_IO1**	I/O	**eSPI Data Signal 1:** Bi-directional pin used to transfer data between the PCH and eSPI device
GPP_A2 / **ESPI_IO2** / SUSWARN# / SUSPWRDNACK	I/O	**eSPI Data Signal 2:** Bi-directional pin used to transfer data between the PCH and eSPI device
GPP_A3 / **ESPI_IO3** / SUSACK#	I/O	**eSPI Data Signal 3:** Bi-directional pin used to transfer data between the PCH and eSPI device
GPP_A4 / **ESPI_CS0#**	O	**eSPI Chip Select 0:** Driving CS# signal low to select eSPI device for the transaction.
GPP_A9 / **ESPI_CLK**	O	**eSPI Clock:** eSPI clock output from the PCH to eSPI device.
GPP_A10 / **ESPI_RESET#**	O	**eSPI Reset:** Reset signal from the PCH to eSPI device.
GPP_A23 / **ESPI_CS1#**	O	**eSPI Chip Select 1 :**Driving CS# signal low to select eSPI slave for the transaction.
GPP_A5 / **ESPI_ALERT0#**	I	**eSPI Alert 0 :**Alert signal from eSPI slave to the PCH. *Note:* If only a single Slave is connected, the eSPI Compatibility Specification requires that the Slave must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.
GPP_A6 / **ESPI_ALERT1#**	I	**eSPI Alert 1 :** Alert signal from eSPI slave to the PCH. *Note:* If only a single Slave is connected, the eSPI Compatibility Spec requires that the Slave must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.

15.3 Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value
ESPI_IO[3:0]	Pull-up	20 kohm +/- 30%
ESPI_CLK	Pull-down	20 kohm+/- 30%
ESPI_ CS [1:0]#	Pull-up	20 kohm +/- 30%
ESPI_ALERT#	Pull-up	15 - 40 kohm

15.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
ESPI_IO [3:0]	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up	OFF
ESPI_CLK	Primary	Internal Pull- down	Driven Low	Driven Low	OFF
ESPI_ CS [1:0] #	Primary	Internal Pull-up	Driven High	Driven High	OFF
ESPI_ALERT#	Primary	Internal Pull-up	Driven High	Driven High	OFF
ESPI_RESET#	Primary	Driven Low	Driven High	Driven High	OFF
Note: Reset reference for primary well pins is RSMRST#.					

16.0 General Purpose Input and Output

The PCH General Purpose Input/Output (GPIO) signals are grouped into multiple groups (such as GPP_A, GPP_B, and so on). All GPIO groups are powered by the PCH Primary well, except for GPD group which is powered by the PCH Deep Sleep well.

The high level features of GPIO:

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

Table 30. 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

16.1 Signal Description

For GPIO pin implementation including multiplexed native functions, default values, signal states, and other characteristics, download the pdf, click on the navigation pane and refer the spreadsheet, **763122_001_GPIO.xlsx**.

16.2 Functional Description

This section provides information on the following topics:

- Configurable GPIO Voltage
- GPIO Buffer Impedance Compensation
- Interrupt / IRQ via GPIO Requirement
- Programmable Hardware Debouncer
- Integrated Pull-ups and Pull-downs
- SCI / SMI# and NMI
- Timed GPIO (TIME_SYNC)

- GPIO Blink (BK) and Serial Blink (SBK)
- GPIO Ownership
- Native Function and TERM Bit Setting

16.2.1 Configurable GPIO Voltage

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

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

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

WARNING

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

NOTES

1. GPIO S group supports 1.8 V only.
2. GPIO R group supports per-group voltage configuration (3.3 V or 1.8 V) only.
3. GPD group supports 3.3 V only.

16.2.2 GPIO Buffer Impedance Compensation

All GPIO buffers require impedance compensation for 1.8 V and 3.3 V operation. The impedance compensation is done via the GPP_RCOMP signal, which requires a precision pull down resistor of 200 Ohm (1%) to GND. Without proper impedance compensation, the GPIO buffers, including the muxed native functions, may not operate as expected.

16.2.3 Interrupt / IRQ via GPIO Requirement

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

16.2.4 Programmable Hardware Debouncer

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

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

16.2.5 Integrated Pull-ups and Pull-downs

All GPIOs have programmable internal pull-up/pull-down resistors which are off by default. The internal pull-up/pull-down for each GPIO can be enabled by BIOS programming the corresponding PAD_CFG_DW1 register. Refer to Volume 2 (Register Information) for more details.

16.2.6 SCI / SMI# and NMI

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

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

- GPP_B14 and GPP_B23
- GPP_C[23:22]
- GPP_D[4:0]
- GPP_E[8:0] ; GPP_E[16:13]
- GPP_F12

16.2.7 Timed GPIO

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

Timed GPIO can be an input or an output:

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

NOTE

TIME_SYNC can be set as input when both Direction (DIR) bit and Enable (EN) bit in Timed GPIO Control Register are set to 1 (refer to Datasheet Volume 2 for the register info). When EN bit is set to 0, TIME_SYNC will default to output low regardless of DIR bit setting.

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

NOTE

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

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

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

16.2.9 GPIO Ownership

Any PCH GPIO can be owned by the host, the Intel® CSME, or ISH depending on how the pin ownership being programmed. The programmed agent will then own the pin exclusively. For example, when a GPIO pad ownership is programmed to the Intel CSME or ISH, the host software no longer has access to the pin programming.

16.2.10 Native Function and TERM Bit Setting

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

Table 31. Native Function Signals Supporting Dynamic Termination Override

Native Function	Signal With Dynamic Termination Override
Intel®HD Audio	HDA_SDI[0:1], HDA_SDO, HDA_SYNC, HDACPU_SDI, HDACPU_SDO, I2S[5:0]_SCLK, I2S[5:0]_SFRM, I2S[5:0]_RXD, DMIC_DATA[1:0], SNDW[3:0]_DATA
Power Management	ACPRESENT, LAN_WAKE#
Touch Host Controller (THC)	THC0_SPI1_IO[3:0], THC0_SPI2_IO[3:0] THC1_SPI2_IO[3:0]

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

The PCH implements seven I^2C controllers for seven independent I^2C interfaces, I2C0-I2C5 , I2C7. Each interface is a two-wire serial interface consisting of a serial data line (SDA) and a serial clock (SCL).

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

The I^2C interfaces support the following features:

- Speed: standard mode (up to 100 Kb/s), fast mode (up to 400 Kb/s), fast mode plus (up to 1 MB/s) and High speed mode (up to 3.2 Mb/s).
- 1.8 V or 3.3 V support (depending on the voltage supplied to the I^2C signal group)
- Master I^2C operation only
- 7-bit or 10-bit addressing
- 7-bit or 10-bit combined format transfers
- Bulk transmit mode
- Ignoring CBUS addresses (an older ancestor of I^2C used to share the I^2C bus)
- Interrupt or polled-mode operation
- Bit and byte waiting at all bus speed
- Component parameters for configurable software driver support
- Programmable SDA hold time (tHD; DAT)
- DMA support with 64-byte DMA FIFO per channel (up to 32-byte burst)
- 64-byte Tx FIFO and 64-byte Rx FIFO
- SW controlled serial data line (SDA) and serial clock (SCL)

NOTES

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

Table 32. Acronyms

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

Table 33. References

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

17.1 Functional Description

This section provides information on the following topics:

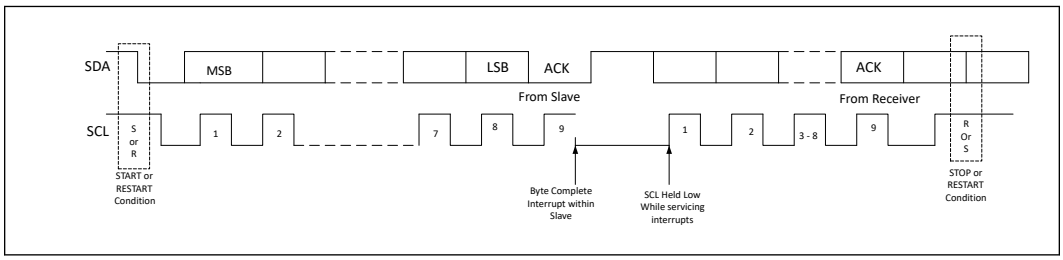
- Protocols overview
- DMA controller
- Reset
- Power Management
- Interrupts
- Error Handling
- Programmable SDA Hold Time

17.1.1 Protocols Overview

For more information on the I²C protocols and command formats, refer to the industry I²C specification. Below is a simplified description of I²C bus operation:

- The master generates a START condition, signaling all devices on the bus to listen for data.
- The master writes a 7-bit address, followed by a read/write bit to select the target device and to define whether it is a transmitter or a receiver.
- The target device sends an acknowledge bit over the bus. The master must read this bit to determine whether the addressed target device is on the bus.
- Depending on the value of the read/write bit, any number of 8-bit messages can be transmitted or received by the master. These messages are specific to the I²C device used. After 8 message bits are written to the bus, the transmitter will receive an acknowledge bit. This message and acknowledge transfer continues until the entire message is transmitted.
- The message is terminated by the master with a STOP condition. This frees the bus for the next master to begin communications. When the bus is free, both data and clock lines are high.

Figure 9. Data Transfer on the I^2C Bus



Combined Formats

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

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

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

17.1.2 DMA Controller

The I^2C controllers 0 to 3 (I2C0 - I2C3) each has an integrated DMA controller. The I2C controller 4 and 5 (I2C4 and I2C5) only implement the I2C host controllers and do not incorporate a DMA. Therefore, I2C4 and I2C5 are restricted to operate in PIO mode only.

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.

Channel Control

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

17.1.3 Reset

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

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

NOTE

To avoid a potential I²C peripheral deadlock condition where the reset goes active in the middle of a transaction, the I²C controller must be idle before a reset can be initiated.

17.1.4 Power Management

Device Power Down Support

To power down peripherals connected to PCH I²C 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 I²C bus is powered off (power gated). The PCH HW will prevent any transitions on the serial bus signals during a power gate event.

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.

17.1.5 Interrupts

I²C 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.

17.1.6 Error Handling

Errors that might occur on the external I²C signals are comprehended by the I²C host controller and reported to the I²C bus driver through the MMIO registers.

17.1.7 Programmable SDA Hold Time

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

17.2 Signal Description

Name	Type	Description
GPP_H4 / **I2C0_SDA**	I/OD	**I²C Link 0 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_H5 / **I2C0_SCL**	I/OD	**I²C Link 0 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_E12 / THC0_SPI1_IO1 / **I2C0A_SDA** / GSPI0_MISO	I/OD	**I²C Link 0A Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance. Note : Alternate interface from/to the same I2C0 controller, to support touch device interface convergence.
GPP_E13 / THC0_SPI1_IO0 / **I2C0A_SCL** / GSPI0_MOSI	I/OD	**I²C Link 0A Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance. Note : Alternate interface from/to the same I2C0 controller, to support touch device interface convergence.
GPP_H6 / **I2C1_SDA**	I/OD	**I²C Link 1 Serial Data Line**

continued...

Name	Type	Description
		External Pull-up resistor may be required depending on Bus Capacitance.
GPP_H7 / **I2C1_SCL**	I/OD	**I²C Link 1 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_F13 / GSXSLOAD / THC1_SPI2_IO1 / GSPI1_MISIO / **I2C1A_SDA**	I/OD	**I²C Link 1A Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance. Note : Alternate interface from/to the same I2C1 controller, to support touch device interface convergence.
GPP_F12 / GSXDOUT / THC1_SPI2_IO0 / GSPI1_MOSI / **I2C1A_SCL**	I/OD	**I²C Link 1A Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance. Note : Alternate interface from/to the same I2C1 controller, to support touch device interface convergence.
GPP_B5 / ISH_I2C0_SDA / **I2C2_SDA**	I/OD	**I²C Link 2 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_B6 / ISH_I2C0_SCL / **I2C2_SCL**	I/OD	**I²C Link 2 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_B7 / ISH_I2C1_SDA / **I2C3_SDA**	I/OD	**I²C Link 3 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_B8 / ISH_I2C1_SCL / **I2C3_SCL**	I/OD	**I²C Link 3 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_H8 / **I2C4_SDA** / CNV_MFUART2_RXD	I/OD	**I²C Link 4 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_H9 / **I2C4_SCL** / CNV_MFUART2_TXD	I/OD	**I²C Link 4 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_B16 / **I2C5_SDA** / ISH_I2C2_SDA	I/OD	**I²C Link 5 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_B17 / **I2C5_SCL** / ISH_I2C2_SCL	I/OD	**I²C Link 5 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_D13 / ISH_UART0_RXD / **I2C6_SDA**	I/OD	**I²C Link 6 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_D14 / ISH_UART0_TXD / **I2C6_SCL**	I/OD	**I²C Link 6 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_H12 / **I2C7_SDA** / UART0_RTS# / SATA_DEVSLP0#	I/OD	**I²C Link 7 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_H13 / **I2C7_SCL** / UART0_CTS# / SATA_DEVSLP1#	I/OD	**I²C Link 7 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_D15 / ISH_UART0_RTS# / **I2C7B_SDA**	I/OD	**2nd instance of the I²C Link 7 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_D16 / ISH_UART0_CTS# / **I2C7B_SCL**	I/OD	**2nd instance of the I²C Link 7 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.

17.3 Integrated Pull-Ups and Pull-Downs

None.

17.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
I2C[6:0]_SDA	Primary	Undriven	Undriven	Undriven	OFF
I2C[6:0]_SCL, I2C7_SCL , I2C[1:0]A_SCL , I2C7B_SCL	Primary	Undriven	Undriven	Undriven	OFF

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

18.0 Gigabit Ethernet Controller

The Gigabit Ethernet controller in conjunction with the Intel® Ethernet Connection I219 provides a complete LAN solution. This chapter describes the behavior of the Gigabit Ethernet Controller.

Table 34. Acronyms

Acronyms	Description
GbE	Gigabit Ethernet

Table 35. 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 / I225 Datasheet	http://www.intel.com/content/www/us/en/

18.1 Functional Description

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

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

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

NOTE

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

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

NOTE

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

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

The integrated GbE controller features are:

- Network Features
  - Compliant with the 1 GB/s Ethernet 802.3, 802.3u, 802.3ab specifications
  - Multi-speed operation: 10/100/1000 Mbps
  - Full-duplex operation at 10/100/1000 Mbps: Half-duplex at 10/100 Mbps
  - Flow control support compliant with the 802.3X specification
  - VLAN support compliant with the 802.3q specification
  - MAC address filters: perfect match unicast filters; multicast hash filtering, broadcast filter and promiscuous mode
  - PCI Express*/SMLink interface to GbE PHYs
- Host Interface Features
  - 64-bit address master support for systems using more than 4 GB of physical memory
  - Programmable host memory receive buffers (256 bytes to 16 KB)
  - Intelligent interrupt generation features to enhance driver performance
  - Descriptor ring management hardware for transmit and receive
  - Software controlled reset (resets everything except the configuration space)
  - Message Signaled Interrupts
- Performance Features
  - Configurable receive and transmit data FIFO, programmable in 1 KB increments
  - TCP segmentation off loading features
  - Fragmented UDP checksum off load for packet reassembly
  - IPv4 and IPv6 checksum off load support (receive, transmit, and large send)
  - Split header support to eliminate payload copy from user space to host space
  - Receive Side Scaling (RSS) with two hardware receive queues
  - Supports 9018 bytes of jumbo packets
  - Packet buffer size 32 KB
  - TimeSync off load compliant with 802.1as specification
  - Platform time synchronization
- Power Management Features

- Magic Packet* wake-up enable with unique MAC address
- ACPI register set and power down functionality supporting D0 and D3 states
- Full wake up support (APM, ACPI)
- MAC power down at Sx, DM-Off with and without WoL
- Auto connect battery saver at S0 no link and Sx no link
- Energy Efficient Ethernet (EEE) support
- Latency Tolerance Reporting (LTR)
- ARP and ND proxy support through LAN Connected Device proxy
- Wake on LAN (WoL) from Deep Sx
- Windows* InstantGo* Support

18.1.1 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.

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.

Data Alignment

- **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.

- **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.

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.

18.1.2 Error Events and Error Reporting

Complete 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.

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.

18.1.3 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.

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 below table.

Table 36. LAN Mode Support

Mode	System State	Interface Active	Connections
Normal 10/100/1000 Mbps	S0	PCI Express*	Intel® Ethernet Connection I219
Normal 10/100/1000 Mbps	S0ix	SMLink0	
Wake-on-LAN	S0ix / Sx / Deep Sx	SMLink0 / LAN_Wake#	
Manageability	S0ix / Sx	SMLink0	

18.1.4 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 (S5) and Deep Sx to S0.

NOTE

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

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

NOTE

Refer to SLP_LAN# Pin Behavior on page 130.

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

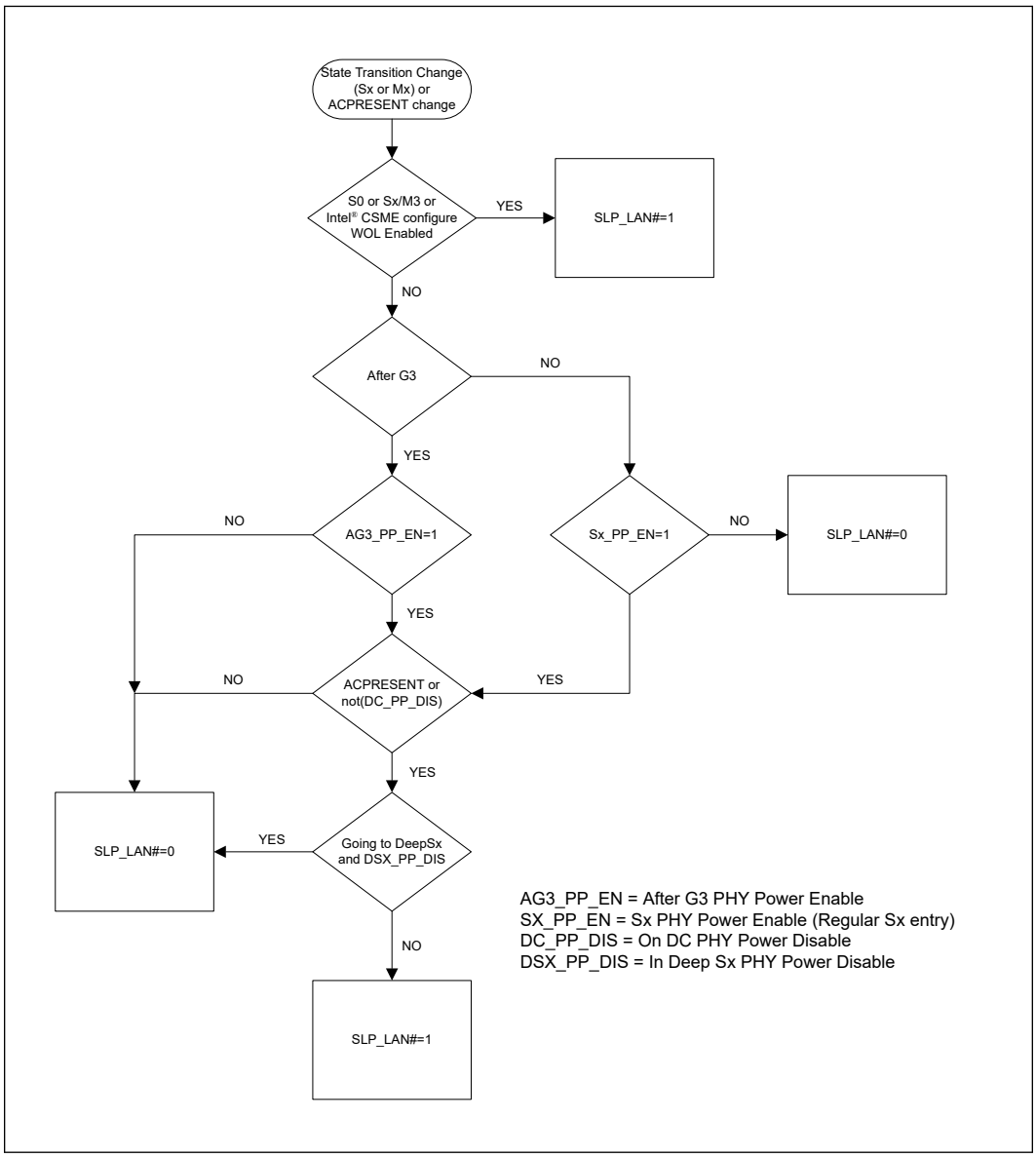
- The LAN PHY is always powered when the Host and Intel® CSME systems are running.
  - SLP_LAN#='1' whenever SLP_S3#='1' or SLP_A#='1'.
- If the LAN PHY is required by Intel® CSME in Sx/M-Off or Deep Sx, Intel® CSME must configure SLP_LAN#='1' irrespective of the power source and the destination power state. Intel® CSME must be powered at least once after G3 to configure this.
- If the LAN PHY is required after a G3 transition, the host BIOS must set AG3_PP_EN.
- If the LAN PHY is required in Sx/M-Off, the host BIOS must set SX_PP_EN.
- If the LAN PHY is required in Deep Sx, the host BIOS must keep DSX_PP_DIS cleared.
- If the LAN PHY is not required if the source of power is battery, the host BIOS must set DC_PP_DIS.

NOTE

Intel® CSME configuration of SLP_LAN# in Sx/M-Off and Deep Sx is dependent on Intel® CSME power policy configuration.

The flow chart below shows how a decision is made to drive SLP_LAN# every time its policy needs to be evaluated.

Figure 10. Conceptual Diagram of SLP_LAN#



18.2 Signal Description

Table 37. GbE LAN Signals

Signal Name	Type	Description
PCIE7_TXP **PCIE7_TXN** **PCIE8_TXP** **PCIE8_TXN** **PCIE9_TXP** **PCIE9_TXN**	O	Differential transmit pairs to the Intel® Ethernet Connection I219 based on the PCIe interface. Refer to PCI Express* (PCIe*) for details on the PCI Express*transmit signals. *Note:* The Intel ® Ethernet Connection I219 can be connected to one of the following PCI Express* ports 7, 8 or 9.

continued...

Signal Name	Type	Description
PCIE7_RXP **PCIE7_RXN** **PCIE8_RXP** **PCIE8_RXN** **PCIE9_RXP** **PCIE9_RXN**	I	Differential receive pairs to the Intel® Ethernet Connection I219 based on the PCIe interface. Refer to PCI Express* (PCIe*) for details on the PCI Express* transmit signals. *Note:* The Intel® Ethernet Connection I219 can be connected to one of the following PCI Express* ports7, 8 or 9.
GPP_C4 / **SML0DATA**	I/OD	System Management Link data signal interface to Intel® Ethernet Connection I219. Refer to System Management Interface and SMLink for details on the SML0DATA signal. *Note:* The Intel® Ethernet Connection I219 connects to SML0DATA signal.
GPP_C3 / **SML0CLK**	I/OD	System Management Link data signal interface to Intel® Ethernet Connection I219. Refer to System Management Interface and SMLink for details on the SML0CLK signal. *Note:* The Intel® Ethernet Connection I219 connects to SML0CLK signal.
GPD11 / **LANPHYPC**	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.
SLP_LAN#	IO	**LAN Sub-System Sleep Control:** If the Gigabit Ethernet Controller is enabled, when SLP_LAN# is de-asserted it indicates that the PHY device must be powered. When SLP_LAN# is asserted, power can be shut off to the PHY device. *Note:* If Gigabit Ethernet Controller is statically disabled via BIOS, SLP_LAN# will be driven low.
GPD2 / **LAN_WAKE#**	I	**LAN WAKE:** LAN Wake Indicator from the GbE PHY. *Note:* LAN_WAKE# functionality is only supported with Intel PHY I219. Connection of a third party LAN device's wake signal to LAN_WAKE# is not supported.

18.3 Integrated Pull-Ups and Pull-Downs

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

18.4 I/O Signal Planes and States

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

Notes: 1. Based on wake events and Intel ME state
2. Configurable based on BIOS settings: '0' When LAN controller is configured as "Disabled" in BIOS, SLP_LAN# will drive "Low";'1' When LAN controller is configured as "Enabled" in BIOS, SLP_LAN# will drive "High"
3. Reset reference for DSW well pins is DSW_PWROK.

19.0 Integrated Sensor Hub (ISH)

Table 38. Acronyms

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

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

19.1 Functional Description

This section provides the information about ISH Micro-Controller, SRAM, PCI Host Interface, Power Domains and Management, ISH IPC and ISH Interrupt Handling via IOAPIC (Interrupt Controller).

19.1.1 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.

19.1.2 SRAM

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

19.1.3 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.

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 Configuration space. It is used only in ACPI mode (that is, when the PCI configuration space is hidden).

DMA Controller

The DMA controller supports up to 64-bit addressing.

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.

PCI Power Management

PME is not supported in ISH.

19.1.4 Power Domains and Management

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 inS4, 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 S5 state.

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

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.

19.1.5 ISH IPC

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

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

Table 39. IPC Initiator -> Target flows

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

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

19.1.6 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.

19.2 Feature Overview

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

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

The ISH consists of the following key components:

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

19.2.1 ISH I²C Controllers

The ISH supports three I²C controllers capable of operating at speeds up to 2.4 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:

- Master Mode Only (all peripherals must be slave devices)
- Support for the following operating speeds:
  - Standard mode: 100 kbps
  - Fast Mode: 400 kbps
  - Fast Mode Plus: 1 000 kbps
  - High Speed Mode: 2400 kbps
- Support for both 7-bit and 10-bit addressing formats on the I²C bus
- FIFO of 64 bytes with programmable watermarks/thresholds

19.2.2 ISH UART Controller

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

The UART has the following capabilities:

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

19.2.3 ISH GSPI Controller

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

The SPI controller includes:

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

19.2.4 ISH GPIOs

The ISH supports eight dedicated GPIOs.

19.3 Signal Description

Signal Name	Type	Description
GPP_B5 / **ISH_I2C0_SDA** / I2C2_SDA	I/OD	ISH I²C 0 Data
GPP_B6 / **ISH_I2C0_SCL** / I2C2_SCL	I/OD	ISH I²C 0 Clk
GPP_B7 / **ISH_I2C1_SDA** / I2C3_SDA	I/OD	ISH I²C 1 Data
GPP_B8 / **ISH_I2C1_SCL** / I2C3_SCL	I/OD	ISH I²C 1 Clk
GPP_B16 / I2C5_SDA / **ISH_I2C2_SDA**	I/OD	ISH I²C 2 Data
GPP_B17 / I2C5_SCL / **ISH_I2C2_SCL**	I/OD	ISH I²C 2 Clk
GPP_D0 / **ISH_GP0** / BK0 / SBK0	I/O	ISH GPIO 0
GPP_D1 / **ISH_GP1** / BK1 / SBK1	I/O	ISH GPIO 1
GPP_D2 / **ISH_GP2** / BK2 /SBK2	I/O	ISH GPIO 2
GPP_D3 / **ISH_GP3** / BK3 /SBK3	I/O	ISH GPIO 3
GPP_E9 / USB2_OC0# / **ISH_GP4**	I/O	ISH GPIO 4
GPP_A16 / USB_OC3# / **ISH_GP5**	I/O	ISH GPIO 5
GPP_B14 / SPKR / TIME_SYNC1 / SATA_LED# / **ISH_GP6**	I/O	ISH GPIO 6
GPP_B15 / TIME_SYNC0 / **ISH_GP7**	I/O	ISH GPIO 7
GPP_B3 / PROC_GP2 / **ISH_GP4B**	I/O	ISH GPIO 4B
GPP_B4 / PROC_GP3 / **ISH_GP5B**	I/O	ISH GPIO 5B
GPP_H12 / I2C7_SDA/ UART0_RTS# / M2_SKT2_CFG0 / **ISH_GP6B** / DEVSLP0B#	I/O	ISH GPIO 6B
GPP_H13 / I2C7_SCL / M2_SKT2_CFG1 / **ISH_GP7B** / DEVSLP1B#	I/O	ISH GPIO 7B
GPP_D14 / **ISH_UART0_TXD** / I2C4B_SCL	O	ISH UART 0 Transmit Data
GPP_D13 / **ISH_UART0_RXD** / I2C4B_SDA	I	ISH UART 0 Receive Data
GPP_D15 / **ISH_UART0_RTS#**	O	ISH UART 0 Request To Send
GPP_D16 / **ISH_UART0_CTS#**	I	ISH UART 0 Clear to Send
GPP_D18 / UART1_TXD / **ISH_UART1_TXD**	O	ISH UART 1 Transmit Data
GPP_D17 / UART1_RXD / **ISH_UART1_RXD**	I	ISH UART 1 Receive Data
GPP_D9 / **ISH_SPI_CS#** / DDP3_CTRLCLK / TBT_LSX2_TXD / GSPI2_CS0#	O	ISH SPI Chip Select
GPP_D10 / **ISH_SPI_CLK** / DDP3_CTRLDATA / TBT_LSX2_RXD / BSSB_LS2_TX / GSPI2_CLK	O	ISH SPI Clock
GPP_D11 / **ISH_SPI_MISO** / DDP4_CTRLCLK / TBT_LSX3_TXD / BSSB_LS3_RX / GSPI2_MISO	I	ISH SPI MISO
GPP_D12 / **ISH_SPI_MOSI** / DDP4_CTRLDATA / TBT_LSX3_RXD / BSSB_LS3_TX / GSPI2_MOSI	O	ISH SPI MOSI

19.4 Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value
ISH_SPI_MISO	Pull-Down	20 kohm ± 30%

19.5 I/O Signal Planes and States

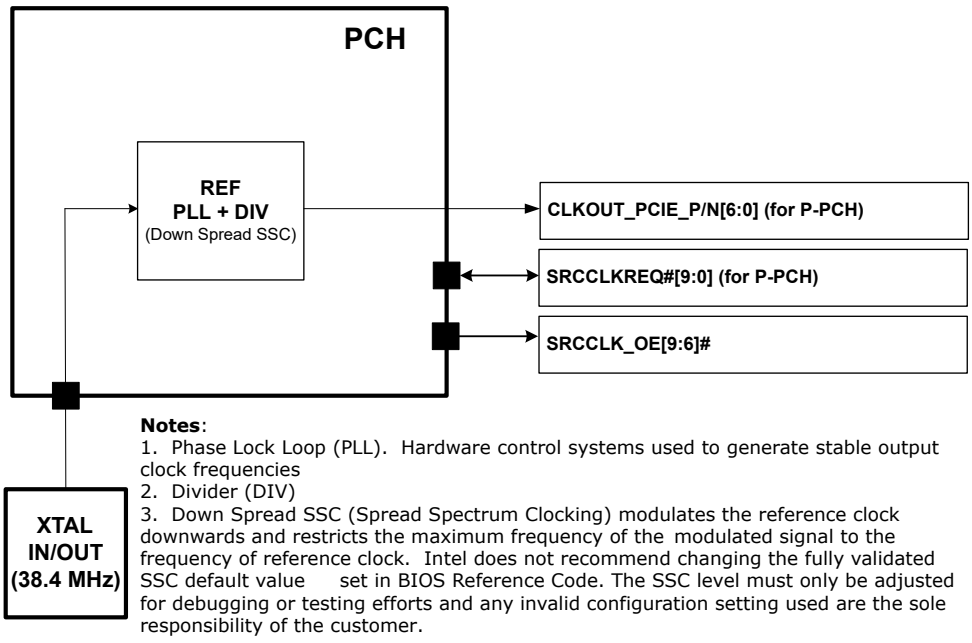
Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
ISH_I2C0_SDA	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C0_SCL	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C1_SDA	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C1_SCL	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C2_SDA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

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

20.0 PCH and System Clocks

20.1 ICC

Figure 11. Integrated Clock Controller (ICC) Diagram



20.1.1 Signal Descriptions

Table 40. Signal Descriptions

Signal Name	Type	SSC Capable	Description
CLKOUT_PCIE_N0 **CLKOUT_PCIE_N1** **CLKOUT_PCIE_N2** **CLKOUT_PCIE_N3** **CLKOUT_PCIE_N4**/UFS_REF_CLK **CLKOUT_PCIE_N5** **CLKOUT_PCIE_N6** (for P-PCH only) **CLKOUT_PCIE_P0** **CLKOUT_PCIE_P1** **CLKOUT_PCIE_P2**	O	Yes	**PCI Express* Clock Output**: Serial Reference 100 MHz PCIe* specification compliant differential output clocks to PCIe* devices.

continued...

Signal Name	Type	SSC Capable	Description
CLKOUT_PCIE_P3 **CLKOUT_PCIE_P4** **CLKOUT_PCIE_P5** **CLKOUT_PCIE_P6** (for P-PCH only)			
GPP_D4/**IMGCLKOUT0**/BK4/SBK4	O		**Imaging Clock** : Clock for external camera sensor.
GPP_H20/**IMGCLKOUT1**	O		**Imaging Clock** : Clock for external camera sensor.
GPP_H21/**IMGCLKOUT2**	O		**Imaging Clock** : Clock for external camera sensor.
GPP_H22/**IMGCLKOUT3**	O		**Imaging Clock** : Clock for external camera sensor.
GPP_D5/**SRCCLKREQ0#** GPP_D6/**SRCCLKREQ1#** GPP_D7/**SRCCLKREQ2#** GPP_D8/**SRCCLKREQ3#** GPP_H19/**SRCCLKREQ4#** GPP_H23/**SRCCLKREQ5#** GPP_F19/**SRCCLKREQ6#** (for P-PCH only) GPP_A8/**SRCCLKREQ7#** (for P-PCH only) GPP_E16/**SRCCLKREQ8#** (for P-PCH only) GPP_E0/SATAXPCIE0/SATAGP0/**SRCCLKREQ9#** (for P-PCH only) GPP_A12/SATAXPCIE1/SATAGP1/**SRCCLKREQ9B#**(for P-PCH only)	I/OD		**Clock Request**: Serial Reference Clock request signals for PCIe* 100 MHz differential clocks.
XTAL_IN	I		**Crystal Input:** Input connection for 38.4 MHz crystal to PCH.
XTAL_OUT	O		**Crystal Output**: Output connection for 38.4 MHz crystal to PCH.
XCLK_BIASREF	I/O		**Differential Clock Bias Reference**: Used to set BIAS reference for differential clocks.
GPP_E5/DEVSLP1/**SRCCLK_OE6#** (for P-PCH only) GPP_A7/**SRCCLK_OE7#** (for P-PCH only) GPP_E15/**SRCCLK_OE8#** (for P-PCH only) GPP_E4/DEVSLP0/**SRCCLK_OE9#** (for P-PCH only)	O		Clock Output Enable for discrete clock buffer.

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 invalid configuration setting used are the sole responsibility of the customer.
2. The SRCCLKREQ# signals can be configured to map to any of the PCH PCI Express* Root Ports while using any of the CLKOUT_PCIE_P/N differential pairs.

20.2 I/O Signal Pin States

Table 41. I/O Signal Pin States

Signal Name	S4/S5	S0 Entry	S0	Deep Sx
CLKOUT_PCIE_P[0:6]	OFF (Gated Low)	Bringing up the Clock	Toggling	OFF (Gated Low)

continued...

Signal Name	S4/S5	S0 Entry	S0	Deep Sx
CLKOUT_PCIE_N[0:6]				
SRCCLKREQ[0:9]#	Un-driven	Un-driven	Un-driven	OFF
SRCCLK_OE[6:9]#	Un-driven	Un-driven	Un-driven	OFF

21.0 PCI Express* (PCIe*)

Table 42. Acronym

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

21.1 Functional Description

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).

21.1.1 Interrupt Generation

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

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

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

Table 43. 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

21.1.2 PCI Express* Power Management

S4/S5 Support

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

Prior to entering , software is required to put each device into D3HOT. When a device is put into D3HOT, it will initiate entry into a L1 link state by sending a PM_Enter_L1 DLLP. Under normal operating conditions when the root ports sends 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.

Device Initiated PM_PME Message

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

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

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

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

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

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 workaround 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.

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**).

21.1.3 Dynamic Link Throttling

Root Port supports dynamic link throttling as a mechanism to help lower the overall component power, ensuring that the component never operates beyond the thermal limit of the package. Dynamic link throttling is also used as a mechanism for ensuring that the ICCmax 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.

21.1.4 Port 8xh Decode

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

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

21.1.5 Separate Reference Clock with Independent SSC (SRIS)

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

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

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

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

21.1.6 Advanced Error Reporting

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

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

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

Alternative Routing - ID Interpretation (ARI)

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

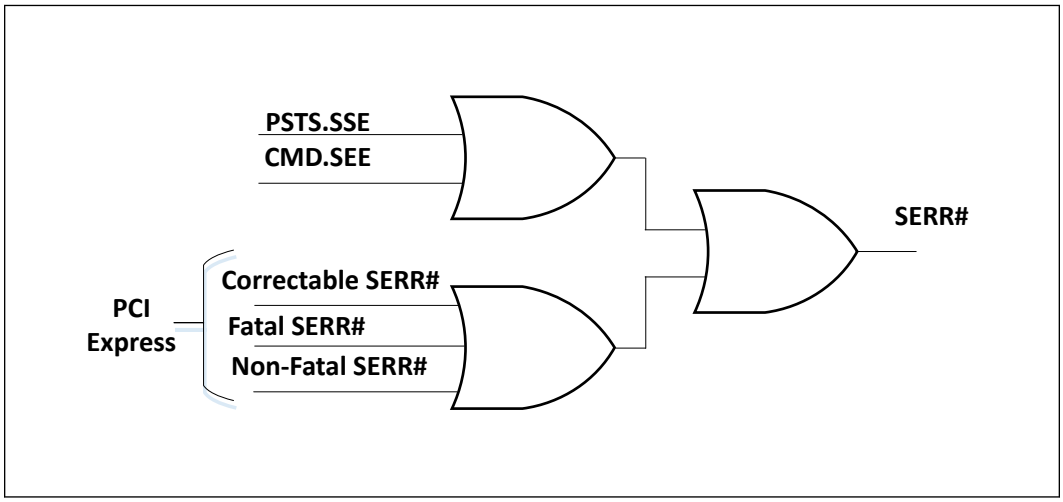
Access Control Services (ACS)

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

21.1.8 SERR# Generation

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

Figure 12. Generation of SERR# to Platform



21.1.9 Hot - Plug

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

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

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.

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 workaround 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.

21.1.10 PCI Express* Lane Polarity Inversion

The PCI Express* Base Specification requires polarity inversion to be supported independently by all receivers across a Link—each differential pair within each Lane of a PCIe* Link handles its own polarity inversion. Polarity inversion is applied, as needed, during the initial training sequence of a Lane. In other words, a Lane will still function correctly even if a positive (Tx+) signal from a transmitter is connected to the negative (Rx-) signal of the receiver. Polarity inversion eliminates the need to untangle a trace route to reverse a signal polarity difference within a differential pair and no special configuration settings are necessary in the PCH to enable it.

NOTE

The 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".

21.1.11 Precision Time Measurement (PTM)

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

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

21.2 Signal Description

PCH	Signal Name	Type	Description
PCH-P	**PCIE1_TXP** / USB32_1_TXP	O	**PCI Express* Differential Transmit Pairs** These are PCI Express* based outbound high-speed differential signals

continued...

PCH	Signal Name	Type	Description
	PCIE1_TXN / USB32_1_TXN **PCIE2_TXP** / USB32_2_TXP **PCIE2_TXN** / USB32_2_TXN **PCIE3_TXP** / USB32_3_TXP **PCIE3_TXN** / USB32_3_TXN **PCIE4_TXP** / USB32_4_TXP **PCIE4_TXN** / USB32_4_TXN **PCIE5_TXP** **PCIE5_TXN** **PCIE6_TXP** **PCIE6_TXN** **PCIE7_TXP** **PCIE7_TXN** **PCIE8_TXP** **PCIE8_TXN** **PCIE9_TXP** / UFS10_TXP **PCIE9_TXN** / UFS10_TXN **PCIE10_TXP** / UFS11_TXP **PCIE10_TXN** / UFS11_TXN **PCIE11_TXP** / SATA0_TXP **PCIE11_TXN** / SATA0_TXN **PCIE12_TXP** / SATA1_TXP **PCIE12_TXN** / SATA1_TXN		
	PCIE1_RXP / USB32_1_RXP **PCIE1_RXN** / USB32_1_RXN **PCIE2_RXP** / USB32_2_RXP **PCIE2_RXN** / USB32_2_RXN **PCIE3_RXP** / USB32_3_RXP **PCIE3_RXN** / USB32_3_RXN **PCIE4_RXP** / USB32_4_RXP **PCIE4_RXN** / USB32_4_RXN **PCIE5_RXP** **PCIE5_RXN** **PCIE6_RXP** **PCIE6_RXN** **PCIE7_RXP** **PCIE7_RXN** **PCIE8_RXP** **PCIE8_RXN**	I	**PCI Express* Differential Receive Pairs** These are PCI Express* based inbound high-speed differential signals

continued...

PCH	Signal Name	Type	Description
	PCIE9_RXP / UFS10_RXP **PCIE9_RXN** / UFS10_RXN **PCIE10_RXP** / UFS11_RXP **PCIE10_RXN** / UFS11_RXN **PCIE11_RXP** / SATA0_RXP **PCIE11_RXN** / SATA0_RXN **PCIE12_RXP** / SATA1_RXP **PCIE12_RXN** / SATA1_RXN		
	PCIE_RCOMPP **PCIE_RCOMPN**	I	**Impedance Compensation Inputs**
	GPP_H15/DDPB_CTRLCLK/ **PCIE_LINK_DOWN**	O	**PCIE_LINK_DOWN Output** PCIe link failure debug signal. PCH PCIe Root Port(s) will assert this signal when a link down event occurs and is detected. For example when a link fails to train during an L1 sub-state exit event.

21.3 I/O Signal Planes and States

Table 44. Power Plane and States for PCI Express* Signals

Signal Name	Type	Power Plane	During Reset[2]	Immediately After Reset[2]	S4/S5	Deep Sx
PCIE_TXP/ PCIE_TXN	O	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
PCIE_RXP/ PCIE_RXN	I	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
PCIE_RCOMPP / PCIE_RCOMPN	I	Primary	Undriven	Undriven	Undriven	OFF

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

21.4 PCI Express* Port Support Feature Details

Table 45. PCI Express* Port Feature Details

PCH	Max Transfer Rate	Max Device (Ports)	Max Lanes	PCIe* Gen Type	Encoding	Transfer Rate (MT/s)	Theoretical Max Bandwidth (GB/s)		
							x1	x2	x4
PCH-P	8 GT/s (Gen3)	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

Notes: 1. Theoretical Maximum Bandwidth (GB/s) = ((Transfer Rate * Encoding * # PCIe Lanes) /8)/1000
- Gen3 Example: = ((8000 * 128/130* 4)/8)/1000 = 3.94 GB/s

2. When GbE is enabled on a PCIe* Root Port, the Max. Device (Ports) value listed is reduced by a factor of 1

Figure 13. PCH-P Supported PCI Express* Link Configurations

PCH-P		PCIe Controller #1				PCIe Controller #2				PCIe Controller #3			
Flex I/O Lanes		0	1	2	3	4	5	6	7	8	9	10	11
PCIe Lanes		1	2	3	4	5	6	7	8	9	10	11	12
Logical Link Lanes	1x4	0	1	2	3	0	1	2	3	0	1	2	3
	1x4 LR	3	2	1	0	3	2	1	0	3	2	1	0
	2x2	0	1	0	1	0	1	0	1	0	1	0	1
	2x2 LR	1	0	1	0	1	0	1	0	1	0	1	0
	1x2+2x1	0	1	0	0	0	1	0	0	0	1	0	0
	2x1+1x2	0	0	1	0	0	0	1	0	0	0	1	0
	4x1	0	0	0	0	0	0	0	0	0	0	0	0
Assigned Root Ports	1x4	RP1				RP5				RP9			
	1x4 LR	RP1				RP5				RP9			
	2x2	RP1		RP3		RP5		RP7		RP9		RP11	
	2x2 LR	RP3		RP1		RP7		RP5		RP11		RP9	
	1x2+2x1	RP1		RP3	RP4	RP5		RP7	RP8	RP9		RP11	RP12
	2x1+1x2	RP4	RP3	RP1		RP8	RP7	RP5		RP12	RP11	RP9	
	4x1	RP1	RP2	RP3	RP4	RP5	RP6	RP7	RP8	RP9	RP10	RP11	RP12

NOTES

1. RP# refers to a specific PCH PCI Express* Root Port #; for example RP3 = PCH PCI Express* Root Port 3
2. A PCIe* Lane is composed of a single pair of Transmit (TX) and Receive (RX) differential pairs, for a total of four data wires per PCIe* Lane (such as, PCIE[3]_TXP/ PCIE[3]_TXN and PCIE[3]_RXP/ PCIE[3]_RXN make up PCIe Lane 3). A connection between two PCIe* devices is known as a PCIe* Link, and is built up from a collection of one or more PCIe* Lanes which make up the width of the link (such as bundling 2 PCIe* Lanes together would make a x2 PCIe* Link). A PCIe* Link is addressed by the lowest number PCIe* Lane it connects to and is known as the PCIe* Root Port (such as a x2 PCIe* Link connected to PCIe* Lanes 3 and 4 would be called x2 PCIe* Root Port 3).
3. The PCIe* Lanes can be configured independently from one another but the max number of configured Root Ports (Devices) must not be exceeded
   - PCH-P: A maximum of 6 PCIe* Root Ports (or devices) can be enabled
     - — When a GbE Port is enabled, the maximum number of PCIe* Ports (or devices) that can be enabled reduces based off the following:
     - — Max PCIe* Ports (or devices) = 6 - GbE (0 or 1)
4. Unidentified lanes within a PCIe* Link Configuration are disabled but their physical lanes are used for the identified Root Port
5. The PCH PCIe* Root Ports can be configured to map to any of the SRCCLKREQ# PCIe* clock request signals and the CLKOUT_PCIE_P/N PCIe* differential clock signal pairs.
6. Lane Reversal Supported Motherboard PCIe* Configurations = 1x4, 2x1+1x2, and 2x2
   - The 2x1+1x2 configuration is enabled by setting the PCIe* Controller soft straps to 1x2+2x1 with Lane Reversal Enabled
   - 1x4 = 1x4 with Lane Reversal Disabled, 1x4 LR = 1x4 with Lane Reversal Enabled
   - 2x2 = 2x2 with Lane Reversal Disabled, 2x2 LR = 2x2 with Lane Reversal Enabled
7. For unused SATA/PCIe* and USB 3.2/PCIe* Combo Lanes, the unused lanes must be statically assigned to PCIe*, SATA, or USB 3.2 via the SATA/PCIe* and USB 3.2/PCIe* Combo Port Soft Straps discussed in the SPI Programming Guide and through the Intel® Flash Image Tool (Intel® FIT) tool.

22.0 Power Management

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_S5# and PLTRST#); configure and respond to wake events; aggregate and report latency tolerance information for devices/peripherals connected to and integrated into the PCH.

NOTE

In this document, Sx refers to S4/S5 states; Deep Sx refers to Deep S4/Deep S5 states.

Table 46. Acronyms

Acronyms	Description
PMC	Power Management Controller
PMIC	Power Management Integrated Circuit
VR	Voltage Regulator

Table 47. References

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

22.1 Functional Description

This section provides information on the following topics:

- Features
- Power Management Sub-state
- PCH and System Power States
- SMI#/SCI Generation
- C-States
- Sleep States
- Event Input Signals and Their Usage
- ALT Access Mode
- System Power Supplies, Planes, and Signals
- Reset Behavior

22.1.1 Features

- Support for *Advanced Configuration and Power Interface (ACPI)* providing power and thermal management
  - ACPI 24-Bit Timer SCI and SMI# Generation
- PCI PME# signal for Wake Up from Low-Power states
- System Sleep State Control
  - ACPI S4 state – Suspend-to-Disk (STD)
  - ACPI G2/S5 state – Soft Off (SOFF)
  - Power Failure Detection and Recovery
  - Deep Sx
- Intel® CSME Power Management Support
  - Wake events from the Intel® CSME (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 conditions.

22.1.2 Power Management States

S0i2.0 and S0i3.0 are the supported power management states. If a platform wants to enable/ disable the S0ix states, BIOS can do by modifying the LPM_EN register.

22.1.3 PCH and System Power States

The table below shows the power states defined for PCH-based platforms. The state names generally match the corresponding ACPI states.

Table 48. 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 States**: C states are processor power states within the S0 system state that provide for various levels of power savings on the processor. The processor manages C states 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/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.

continued...

State / Substates	Legacy Name/Description
S0ix	**S0 Idle States:** Processor PKG C states and platform latency tolerance will allow the PCH to decide when to take aggressive power management actions.
Deep Sx	**Deep Sx:** An optional low power state where system context may or may not be maintained depending upon entry condition. All power is shut off except for minimal logic that allows exiting Deep Sx. If Deep Sx state was entered from S4 state, then the resume path will place system back into S4. If Deep Sx state was entered from S5 state, then the resume path will place system back into S5.
G3	**Mechanical OFF (M-Off):** System context not maintained. All power is shut off except for the RTC. No "Wake" events are possible. This state occurs if the user removes the main system batteries in a mobile system, turns off a mechanical switch, or if the system power supply is at a level that is insufficient to power the "waking" logic. When system power returns, transition will depend on the state just prior to the entry to G3 and the AFTERG3_EN bit in the General Power Management Configuration (GEN_PMCON). Refer to table System Power Plane for more details.

The table below 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 S5, it may appear to pass through the G1/S4 state. These intermediate transitions and states are not listed in the table below.

Table 49. State Transition Rules for the PCH

Present State	Transition Trigger	Next State
G0/S0/C0	• SLP_EN bit set • Power Button Override[3,5] • Mechanical Off/Power Failure	• G0/S0/Cx • G1/S4, or G2/S5 state • G2/S5 • G3
G0/S0/Cx	• Power Button Override[3,5] • Mechanical Off/Power Failure	• G0/S0/C0 • S5 • G3
G1/S4	• Any Enabled Wake Event • Power Button Override[3,5] • Conditions met as described in PCH and System Power States on page 113 • Mechanical Off/Power Failure	• G0/S0/C0[2] • G2/S5 • Deep S4 • G3
G2/S5	• Any Enabled Wake Event • Conditions met as described in PCH and System Power States on page 113 • Mechanical Off/Power Failure	• G0/S0/C0[2] • Deep S5 • G3
G2/Deep Sx	• Any Enabled Wake Event • ACPRESENT Assertion • Mechanical Off/Power Failure • Power Button Override	• G0/S0/C0[2] • G1/S4 or G2/S5 (Refer to PCH and System Power States on page 113) • G3

continued...

Present State	Transition Trigger	Next State
		• G2/S5
G3	• Power Returns	• S0/C0 (reboot) or G2/S5[4] (stay off until power button pressed or other wake event)[1,2]

Notes: 1. Some wake events can be preserved through power failure.
2. Transitions from the S4-S5 states to the S0 state are deferred until BATLOW# is inactive.
3. Includes all other applicable types of events that force the host into and stay in G2/S5.
4. If the system was in G1/S4 before G3 entry, then the system will go to S0/C0 or G1/S4.
5. Upon entry to S5 due to a power button override, if Deep S5 is enabled and conditions are met per section PCH and System Power States on page 113, the system will transition to Deep S5.

System Power Planes

The system has several independent power planes, as described in the table below.

NOTE

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

Table 50. 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 The processor, PCI Express* will typically be power-gated when the Main power plane is shut down, although there may be small subsections powered. *Note:* The PCH power is not controlled by the SLP_S3# signal, but instead by the SLP_SUS# signal.
Memory	SLP_S4# signal SLP_S5# signal	When SLP_S4# goes active, power can be shut off to any circuit not required to wake the system from the S4. Since the memory context does not need to be preserved in the S4 state, the power to the memory can also be shut down. When SLP_S5# goes active, power can be shut off to any circuit not required to wake the system from the S5 state. Since the memory context does not need to be preserved in the S5 state, the power to the memory can also be shut down.
Intel® CSME	SLP_A#	SLP_A# signal is asserted when the Intel® CSME goes to M-Off or M3-PG. Depending on the platform, this pin may be used to control power to various devices that are part of the Intel® CSME sub-system in the platform.
LAN	SLP_LAN#	This signal is asserted in Sx/M-Off or Sx/M3-PG when both host and Intel® CSME WoL are not supported. This signal can be use to control power to the Platform LAN Connect Device.
Primary Well	SLP_SUS#	This signal is asserted when the Primary rails can be externally shut off for enhanced power saving.
VCCIO and VCCSTG	PROC_C10	This signal is asserted when the processor enters C10 and can handle VCCIO, VCCSTG and VCCPLL_OC being lowered to 0 V.
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.

22.1.4 SMI#/SCI Generation

Upon any enabled SMI event taking place while the End of SMI (EOS) bit is set, the PCH will clear the EOS bit and assert SMI to the processor, which will cause it to enter SMM space. SMI assertion is performed using a Virtual Legacy Wire (VLW) message.

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.

The table below 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 51. Causes of SMI and SCI

Cause	SCI	SMI	Additional Enables[1]	Where Reported
PME#	Yes	Yes	PME_EN=1	PME_STS
PME_B0 (Internal, Bus 0, PME-Capable Agents)	Yes	Yes	PME_B0_EN=1	PME_B0_STS
PCI Express* PME Messages	Yes	Yes	PCI_EXP_EN=1 (Not enabled for SMI)	PCI_EXP_STS
PCI Express* Hot-Plug Message	Yes	Yes	HOT_PLUG_EN=1 (Not enabled for SMI)	HOT_PLUG_STS
Power Button Press	Yes	Yes	PWRBTN_EN=1	PWRBTN_STS
Power Button Override (Note 6)	Yes	No	None	PWRBTNOR_STS
RTC Alarm	Yes	Yes	RTC_EN=1	RTC_STS
ACPI Timer overflow (2.34 seconds)	Yes	Yes	TMROF_EN=1	TMROF_STS
GPIO	Yes	Yes	Refer to Note 8	
LAN_WAKE#	Yes	Yes	SCI_EN=0, LAN_WAKE_EN=1	LAN_WAKE_STS
TCO SCI message from processor	Yes	No	None	CPUSCI_STS
TCO SCI Logic	Yes	No	TCOSCI_EN=1	TCOSCI_STS
TCO SMI Logic	No	Yes	TCO_EN=1	TCO_STS
TCO SMI – Year 2000 Rollover	No	Yes	None	NEWCENTURY_STS

continued...

Cause	SCI	SMI	Additional Enables[1]	Where Reported
TCO SMI – TCO TIMEROUT	No	Yes	None	TIMEOUT
TCO SMI – OS writes to TCO_DAT_IN register	No	Yes	None	OS_TCO_SMI
TCO SMI – NMI occurred (and NMIs mapped to SMI)	No	Yes	NMI2SMI_EN=1	TCO_STS, NMI2SMI_STS
TCO SMI – INTRUDER# signal goes active	No	Yes	INTRD_SEL=10	INTRD_DET
TCO SMI – Changes of the WPD (Write Protect Disable) bit from 0 to 1	No	Yes	LE (Lock Enable)=1	BIOSWR_STS
TCO SMI – Write attempted to BIOS	No	Yes	WPD=0	BIOSWR_STS
BIOS_RLS written to 1 (Note 7)	Yes	No	GBL_EN=1	GBL_STS
GBL_RLS written to	No	Yes	BIOS_EN=1	BIOS_STS
Write to B2h register	No	Yes	APMC_EN = 1	APM_STS
Periodic timer expires	No	Yes	PERIODIC_EN=1	PERIODIC_STS
64 ms timer expires	No	Yes	SWSMI_TMR_EN=1	SWSMI_TMR_STS
Enhanced USB Legacy Support Event	No	Yes	LEGACY_USB2_EN = 1	LEGACY_USB2_STS
Serial IRQ SMI reported	No	Yes	None	SERIRQ_SMI_STS
Device monitors match address in its range	No	Yes	Refer to DEVTRAP_STS register description	DEVTRAP_STS
SMBus Host Controller	No	Yes	SMB_SMI_EN, Host Controller Enabled	SMBus host status reg.
SMBus Slave SMI message	No	Yes	None	SMBUS_SMI_STS
SMBus SMBALERT# signal active	No	Yes	None	SMBUS_SMI_STS
SMBus Host Notify message received	No	Yes	HOST_NOTIFY_INTREN	SMBUS_SMI_STS, HOST_NOTIFY_STS
BATLOW# assertion	Yes	Yes	BATLOW_EN=1	BATLOW_STS
Access microcontroller 62h/66h	No	Yes	MCSMI_EN	MCSMI_STS
SLP_EN bit written to 1	No	Yes	SMI_ON_SLP_EN=1	SMI_ON_SLP_EN_STS
SPI Command Completed	No	Yes	None	SPI_SMI_STS
eSPI SCI/SMI Request [9]	Yes	Yes	eSPI_SCI_EN	eSPI_SCI_STS eSPI_SMI_STS
Software Generated GPE	Yes	Yes	SWGPE_EN=1	SWGPE_STS
Intel® CSME	Yes	Yes	CSME_SCI_EN=1 CSME_SCI_EN=0; CSME_SMI_EN=1;	CSME_SCI_STS CSME_SMI_STS
GPIO Lockdown Enable bit changes from '1' to '0'	No	Yes	GPIO_UNLOCK_SMI_EN=1	GPIO_UNLOCK_SMI_STS
USB 3.2 (xHCI) SMI Event	No	Yes	xHCI_SMI_EN=1	xHCI_SMI_STS
Wake Alarm Device Timer	Yes	Yes	WADT_EN	WADT_STS
ISH	Yes	No	ISH_EN	ISH_STS
RTC update-in-progress	No	Yes	Refer to Datasheet Volume 2	RTC_UIP_SMI_STS

continued...

Cause	SCI	SMI	Additional Enables[1]	Where Reported
SIO SMI events	No	Yes	SIP_SMI_EN	SIO_SMI_STS
SCC	No	Yes	SCC_SMI_EN	SCC_SMI_STS

Notes:
1. SCI_EN must be 1 to enable SCI, except for BIOS_RLS. SCI_EN must be 0 to enable SMI.
2. SCI can be routed to cause interrupt 9:11 or 20:23 (20:23 only available in APIC mode).
3. GBL_SMI_EN must be 1 to enable SMI.
4. EOS must be written to 1 to re-enable SMI for the next 1.
5. The PCH must have SMI fully enabled when the PCH is also enabled to trap cycles. If SMI is not enabled in conjunction with the trap enabling, then hardware behavior is undefined.
6. When a power button override first occurs, the system will transition immediately to S5. The SCI will only occur after the next wake to S0 if the residual status bit (PRBTNOR_STS) is not cleared prior to setting SCI_EN.
7. GBL_STS being set will cause an SCI, even if the SCI_EN bit is not set. Software must take great care not to set the BIOS_RLS bit (which causes GBL_STS to be set) if the SCI handler is not in place.
8. Refer to General Purpose Input and Output on page 74 for specific GPIOs enabled for SCIs and/or SMIs
9. eSPI slave must assert SCI at least 100 us for the SCI event to be recognized.

PCI Express* SCI

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

PCI Express* Hot-Plug

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

22.1.5 C-States

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

22.1.6 Sleep States

Sleep State Overview

The PCH supports different sleep states (S4/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.

Initiating Sleep State

Sleep states (S4/S5) are initiated by:

- Masking interrupts, turning off all bus master enable bits, setting the desired type in the SLP_TYP field, and then setting the SLP_EN bit. The hardware then attempts to gracefully put the system into the corresponding Sleep state.

- Pressing the PWRBTN# Signal for more than 4 seconds to cause a Power Button Override event. In this case the transition to the S5 state is less graceful, since there are no dependencies 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 the S0 state.
- Shutdown by integrated manageability functions (ASF/Intel® CSME).
- Internal watchdog timer timeout events.

Table 52. Sleep Types

Sleep Type	Comment
S4	The PCH asserts SLP_S3# and SLP_S4#. The motherboard uses the SLP_S4# signal to shut off the power to the memory subsystem and any other unneeded subsystem. Only devices needed to wake from this state should be powered.
S5	The PCH asserts SLP_S3#, SLP_S4# and SLP_S5#.

Exiting Sleep States

Sleep states (S4/S5) are exited based on wake events. The wake events forces the system to a full on state (S0), although some non-critical subsystems might still be shut off and have to be brought back manually. For example, the storage subsystem 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 the table below.

NOTE

If the BATLOW# signal is asserted, the PCH does not attempt to wake from an S4/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 53. Causes of Wake Events

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

continued...

Cause	How Enabled	Wake from Sx	Wake from Deep Sx	Wake from Sx After Power Loss[2]	Wake from "Reset" Types[3]
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	PCIEXP_WAKE_DIS bit.	Yes	Yes	Yes	No
SMBALERT#	Refer to Note 4	Yes	No	Yes	Yes
SMBus Slave Wake Message (01h)	Wake/SMI# command always enabled as a Wake event. *Note:* SMBus Slave Message can wake the system from S4/S5, as well as from S5 due to Power Button Override.	Yes	No	Yes	Yes
SMBus Host Notify message received	HOST_NOTIFY_WKEN bit SMBus Slave Command register. Reported in the SMB_WAK_STS bit in the GPE0_STS register.	Yes	No	Yes	Yes
Intel® CSME Non-Maskable Wake	Always enabled as a wake event.	Yes	No	Yes	Yes
Integrated WoL Enable Override	WoL Enable Override bit (in Configuration Space).	Yes	Yes	Yes	Yes
Wake Alarm Device	WADT_EN in GPE0_EN[127:96]	Yes	Yes	No	No
AC_PRESENT	ACPRESENT_WAKE_EN (Note 6)	No	Yes	No	No
USB connection in/after Deep Sx	GPE0_EN.USB_CON_DSX_EN+	Refer to Note 7	Yes	No	No

Notes:

1. If BATLOW# signal is low, PCH will not attempt to wake from S4/S5 (nor will it exit Deep Sx), even if a valid wake event occurs. This prevents the system from waking when battery power is insufficient to wake the system. However, once BATLOW# de-asserts, the system will boot.
2. This column represents what the PCH would honor as wake events but there may be enabling dependencies on the device side which are not enabled after a power loss.
3. Reset Types include: Power Button override, Intel® CSME-initiated power button override, Intel® CSME-initiated host partition reset with power down, Intel® CSME Watchdog Timer, SMBus unconditional power down, processor thermal trip, PCH catastrophic temperature event.
4. SMBALERT# signal is multiplexed with a GPIO pin that defaults to GPIO mode. Hence, SMBALERT# related wakes are possible only when this GPIO is configured in native mode, which means that BIOS must program this GPIO to operate in native mode before this wake is possible. Because GPIO configuration is in the resume well, wakes remain possible until one of the following occurs: BIOS changes the pin to GPIO mode, a G3 occurs or Deep Sx entry occurs.
5. There are only 72 bits in the GPE registers to be assigned to GPIOs, though any of the GPIOs can trigger a wake, only those status of GPIO mapped to 1-tier scheme are directly accessible through the GPE status registers. For those GPIO mapped under 2-tier scheme, their status would be reflected under single master status, "GPIO_TIER2_SCI_STS" or GPE0_STS and further comparison needed to know which 2-tier GPI(s) has triggered the GPIO Tier 2 SCI.
6. A change in ACPRESENT causes an exit from Deep Sx to Sx, but the system will not wake all the way to S0.
7. Connection of a USB device can cause a wake from normal Sx as well. But that class of wakes is routed through PME_B0, not through this wake enable. The USB_CON_DSX_EN applies only to connection wakes while in Deep Sx or while in Sx after Deep Sx. Note: Sx after Deep Sx reached due to an Intel® CSME wake from Deep Sx or due to ACPRESENT going high while in Deep Sx if Deep Sx is only enabled while on DC power. The following additional conditions are required for this wake to occur:
   - The bit(s) in PM_CFG2.USB_DSX_PER_PORT_EN associated with the port(s) which experienced the connection must be set to '1'.
   - DSX_CFG.USB_CON_DSX_MODE must be set to '1', routing USB connection to generate a wake rather than be reflected out to a pin

PCI Express* WAKE# Signal and PME Event Message

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

NOTE

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

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

Sx-G3-Sx, Handling Power Failures

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

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

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

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

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

Table 54. Transitions Due to Power Failure

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

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

Deep Sx

To minimize power consumption while in S4/S5, the PCH supports a lower power, lower featured version of these power states known as Deep Sx. In the Deep Sx state, the primary 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.

- **Entry Into Deep Sx**

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

- RTCPMCFG.INT_SUS_PD_EN = 1

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

OR

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

Table 55. Supported Deep Sx Policy Configurations

Configuration	S4DC_GATE_SUS	S4AC_GATE_SUS	S5DC_GATE_SUS	S5AC_GATE_SUS
1. Enabled in S5 Battery Only (ACPRESENT = 0)	0	0	1	0
1. Enabled in S5 (ACPRESENT not considered)	0	0	1	1
1. Enabled in S4 and S5 when on Battery only (ACPRESENT = 0)	1	0	1	0
1. Enabled in S4 and S5 (ACPRESENT not considered)	1	1	1	1
1. Enabled in S4, and S5 when on Battery only (ACPRESENT = 0)	1	0	1	0

continued...

Configuration	S4DC_GATE_SUS	S4AC_GATE_SUS	S5DC_GATE_SUS	S5AC_GATE_SUS
1. Enabled in S4, and S5 (ACPRESENT not considered)	1	1	1	1
1. Deep S4 / S5 disabled	0	0	0	0
Note: All other configurations are RESERVED.				

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

- **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 primary well by de-asserting SLP_SUS#.

Table 56. Deep Sx Wake Events

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

ACPRESENT has some behaviors that are different from the other Deep Sx wake events. If the Intel® CSME has enabled ACPRESENT as a wake event then it behaves just like any other Intel® CSME Deep Sx wake event. However, even if ACPRESENT wakes are not enabled, if the Host policies indicate that Deep Sx is only supported when on battery, then ACPRESENT going high will cause the PCH to exit Deep Sx. In this case, the primary wells gets powered up and the platform remains in Sx/M-Off or Sx/M3-PGS3/M-Off. If ACPRESENT subsequently drops (before any Host or Intel® CSME wake events are detected), the PCH will re-enter Deep Sx.

22.1.7 Event Input Signals and Their Usage

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

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 the below table.

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

During the time that any SLP_* signal is stretched for an enabled minimum assertion width, the host wake-up is held off. As a result, it is possible that the user will press and continue to hold the Power Button waiting for the system to wake. Unfortunately, a 4 second press of the Power Button is defined as an unconditional power down, resulting in the opposite behavior that the user was intending. Therefore, the Power Button Override Timer will be extended to 9-10 seconds while the SLP_* stretching timers are in progress. Once the stretching timers have expired, the Power Button will awake the system. If the user continues to press Power Button for the remainder of the 9-10 seconds it will result in the override condition to S5. Extension of the Power Button Override timer is only enforced following graceful sleep entry and during host partition resets with power cycle or power down. The timer is not extended immediately following power restoration after a global reset, G3 or Deep Sx.

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

Table 57. 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]
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

continued...

Present State	Event	Transition/Action	Comment
			Notes: 1. During G3 exit, PWRBTN# pin must be kept de-asserted for a minimum time of 500 us after the RSMRST# has de-asserted.[2] 2. Beyond this point, the minimum time the PWRBTN# pin has to be asserted to be registered by PCH as a valid wake event is 150 us.[1]
S0 – S4	PWRBTN# held low for at least 4 3 consecutive seconds	Unconditional transition to S5 state and if Deep Sx is enabled and conditions are met, the system will then transition to Deep Sx.	No dependence on processor or any other subsystem *Note:* Due to internal PCH latency, it could take up to an additional ~1.3s after PWRBTN# has been held low for 4s before the system would begin transitioning to S5.
Notes: 1. If PM_CFG.PB_DB_MODE='0', the debounce logic adds 16 ms to the start/minimum time for processing of power button assertions. 2. This minimum time is independent of the PM_CFG.PB_DB_MODE value. 3. The amount of time PWRBTN# must be asserted is configurable via PM_CFG2.PBOP. 4 seconds is the default.			

Power Button Override Function

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

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

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

NOTE

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

Sleep Button

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

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

PME# (PCI Power Management Event)

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

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

SYS_RESET# Signal

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

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

NOTES

1. The normal behavior for a SYS_RESET# assertion is host partition reset without power cycle. However, if bit 3 of the CF9h I/O register is set to '1' then SYS_RESET# will result in a full power-cycle reset.
2. It is not recommended to use the PCH_PWROK pin for a reset button as it triggers a global power cycle reset.
3. SYS_RESET# is in the primary power well but it only affects the system when PCH_PWROK is high.

THERMTRIP# Signal

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

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

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

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

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

NOTE

A thermal trip event will clear the PWRBTN_STS bit.

Sx_Exit_Holdoff#

When 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 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 5 ms after SLP_A# assertion before making the determination that it is safe to access flash.

- If no flash activity is seen within this 5 ms 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 5 ms window, the PCH has gone through a global reset. And so the EC must wait until the PCH reaches S0 again before re-attempting the holdoff flow.

NOTE

When eSPI is enabled, the flash sharing functionality using SX_EXIT_HOLDOFF# is not supported, but the pin still functions to hold off Sx exit after SLP_A# de-assertion.

22.1.8 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.

Write Only Registers with Read Paths in ALT Access Mode

The registers described in below table 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 58. Write Only Registers with Read Paths in ALT Access Mode

Restore Data			
I/O Addr	**# of Rds**	**Access**	**Data**
20h	12	1	PIC ICW2 of Master controller
		2	PIC ICW3 of Master controller
		3	PIC ICW4 of Master controller
		4	PIC OCW1 of Master controller [1]
		5	PIC OCW2 of Master controller
		6	PIC OCW3 of Master controller
		7	PIC ICW2 of Slave controller
		8	PIC ICW3 of Slave controller
		9	PIC ICW4 of Slave controller
		10	PIC OCW1 of Slave controller [1]
		11	PIC OCW2 of Slave controller

continued...

Restore Data			
I/O Addr	**# of Rds**	**Access**	**Data**
		12	PIC OCW3 of Slave controller
40h	7	1	Timer Counter 0 status, bits [5:0]
		2	Timer Counter 0 base count low byte
		3	Timer Counter 0 base count high byte
		6	Timer Counter 2 base count low byte
		7	Timer Counter 2 base count high byte
42h	1		Timer Counter 2 status, bits [5:0]
70h	1		Bit 7 = Read value is '0'. Bits [6:0] = RTC Address
Notes: 1. The OCW1 register must be read before entering ALT access mode. 2. Bits 5, 3, 1, and 0 return 0.			

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 below.

Table 59. PIC Reserved Bits Return Values

PIC Reserved Bits	Value Returned
ICW2(2:0)	000
ICW4(7:5)	000
ICW4(3:2)	00
ICW4(0)	0
OCW2(4:3)	00
OCW3(7)	0
OCW3(5)	Reflects bit 6
OCW3(4:3)	01

22.1.9 System Power Supplies, Planes, and Signals

Power Plane Control

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

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

The SLP_S4# output signal is used to remove power to additional subsystems that are powered during SLP_S3#, as well as power to the system memory, since the context of the system is saved on the disk. Cutting power to the memory may be done using the power supply, or by external FETs on the motherboard.

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

SLP_A# output signal can be used to cut power to the Intel® Converged Security and 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® GbE PHY device.

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.

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.

BATLOW# (Battery Low)

The BATLOW# input can inhibit waking from 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.

SLP_LAN# Pin Behavior

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

- The LAN PHY is always powered when the Host and Intel® CSME systems are running.

  - SLP_LAN#='1' whenever SLP_S3#='1' or SLP_A#='1'.

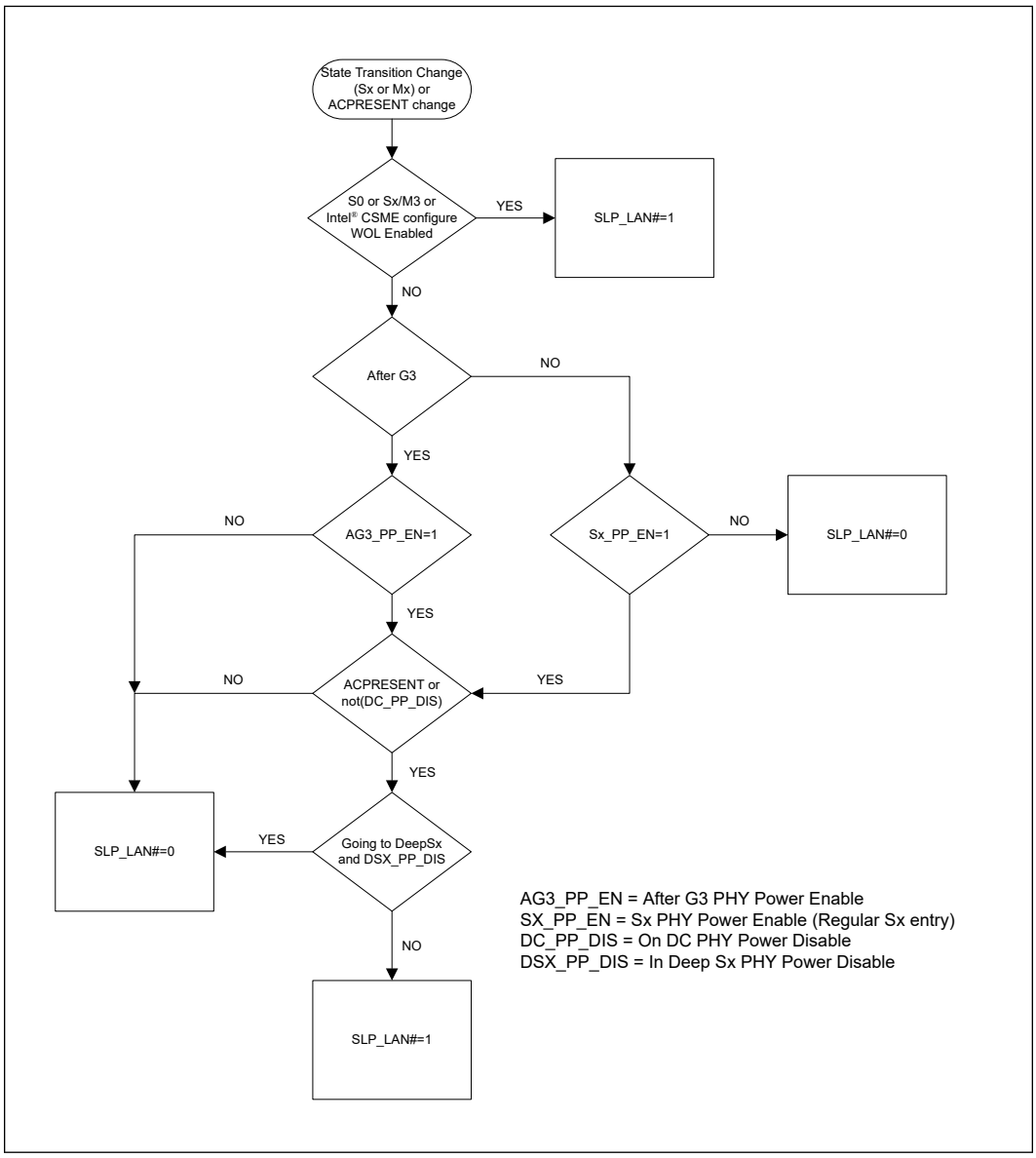
- If the LAN PHY is required by Intel® CSME in Sx/M-Off or Deep Sx, Intel® CSME must configure SLP_LAN#='1' irrespective of the power source and the destination power state. Intel® CSME must be powered at least once after G3 to configure this.
- If the LAN PHY is required after a G3 transition, the host BIOS must set AG3_PP_EN.
- If the LAN PHY is required in Sx/M-Off, the host BIOS must set SX_PP_EN.
- If the LAN PHY is required in Deep Sx, the host BIOS must keep DSX_PP_DIS cleared.
- If the LAN PHY is not required if the source of power is battery, the host BIOS must set DC_PP_DIS.

NOTE

Intel® CSME configuration of SLP_LAN# in Sx/M-Off and Deep Sx is dependent on Intel® CSME power policy configuration.

The flow chart below shows how a decision is made to drive SLP_LAN# every time its policy needs to be evaluated.

Figure 14. Conceptual Diagram of SLP_LAN#



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 S4/S5 states, the host BIOS must set HOST_WLAN_PP_EN.
- If WoWLAN is required from Deep Sx, the host BIOS must set DSX_WLAN_PP_EN.
- If Intel® CSME has access to the Wireless LAN device:

- The Wireless LAN device must always be powered as long as Intel® CSME is powered. SLP_WLAN#='1' whenever SLP_A#='1'.
- If Wake on Wireless LAN (WoWLAN) is required from M-Off state, Intel® CSME will configure SLP_WLAN#='1' in Sx/M-Off.

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

When the Wireless LAN device is an integrated connectivity device (CNVi) the power to the CNVi external RF chip (CRF) must be always on. In this case the SLP_WLAN# shall not control the CRF 3.3 V power rail.

EXT_PWR_GATE# Pin Behavior

EXT_PWR_GATE# can be used to control a FET gating off the HSIO/SRAM power supply to PCH. This provides additional power savings during connected standby states. The ramp time of the FET can be controlled via MODPHY_PM_CFG3.

It is expected that the HSIO/SRAM supply will ramp along with the other primary wells, and must be valid for at least 10 ms before RSMRST# deassertion during a G3/ Deep Sx -> Sx transition. System designers will need to account for this behavior to make sure the rail turns on as expected.

SUSPWRDNACK/SUSWARN#/GPP_A13 Steady State Pin Behavior

Below table summarizes SUSPWRDNACK/SUSWARN#/GPP_A13 pin behavior.

Table 60. 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® CSME power package and power source (Note 1)	0	Off
SUSWARN#	Supported	Native	1	1 (Note 2)	1	Off
GPP_A13	Do not Care	IN	High-Z	High-Z	High-Z	Off
	Do not 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® CSME power policy configuration.
2. If entering Deep Sx, pin will assert and become undriven ("Off") when suspend well drops upon Deep Sx entry.

Table 61. SUSPWRDNACK During Reset

Reset Type (Note)	SPDA Value
Power-cycle Reset	0
Global Reset	0
Straight to S5	PCH initially drive '0' and then drive per Intel® CSME power policy configuration.
Note: Refer to Table 62 on page 136	

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® Converged Security and 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.

PROC_C10_GATE#

When asserted,PROC_C10_GATE# is the indication to the system that the processor is entering C10 and can handle the voltages on the VCCIO, VCCSTG and VCCPLL_OC rails being lowered to 0 V. When de-asserted, the VCCIO and VCCSTG rails must ramp back up to their operational voltage levels. The power good indicators for these rails must still be asserted high when these rails are lowered to 0 V during PROC_C10_GATE# assertion and while these rails ramp back up to their operational levels after PROC_C10_GATE# de-assertion.

NOTE

VCCIO, VCCSTG and VCCPLL_OC are processor power rails.

SLP_S0#

SLP_S0# is the indication to the system to enter the deterministic idle state (S0i2.0 or S0i3.0). This is a PCH hardware controlled output pin. This signal is defined as active low which means a 0 V indicates the deterministic idle state. Additional power saving steps such as VPCLVM may happen during this state.

22.1.10 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.

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.

22.1.11 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 four seconds if an acknowledge from the processor is not received.

When the PCH causes a reset by asserting PLTRST#, its output signals will go to their reset states.

A reset in which the host platform is reset and PLTRST# is asserted is called a Host Reset or Host Partition Reset. Depending on the trigger a host reset may also result in power cycling, refer to the below table for details. If a host reset is triggered and the PCH times out before receiving an acknowledge message from the processor a Global Reset with power-cycle will occur.

A reset in which the host and Intel® CSME partitions of the platform are reset is called a Global Reset. During a Global Reset, all PCH functionality is reset except RTC Power Well backed information and Suspend well status, configuration, and functional logic for controlling and reporting the reset. Intel® CSME and Host power back up after the power-cycle period.

Straight to S5 is another reset type where all power wells that are controlled by the SLP_S3#, SLP_S4#, and SLP_A# pins, as well as SLP_S5# and SLP_LAN# (if pins are not configured as GPIOs), are turned off. All PCH functionality is reset except RTC Power Well backed information and Suspend well status, configuration, and functional logic for controlling and reporting the reset. The host stays there until a valid wake event occurs.

The following table shows the various reset triggers.

Table 62. Causes of Host and Global Resets

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[4]	
Write of 06h to CF9h (RST_CNT Register) when CF9h when Global Reset Bit=0b	Yes	No	No[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[4]	
SYS_RESET# Asserted and CF9h (RST_CNT Register) Bit 3 = 1	No	Yes	No[4]	
SMBus Slave Message received for Reset with Power-Cycle	No	Yes	No[4]	
SMBus Slave Message received for Reset without Power-Cycle	Yes	No	No[4]	
SMBus Slave Message received for unconditional Power Down	No	No	No	Yes
TCO Watchdog Timer reaches zero two times	Yes	No	No[4]	
Power Failure: PCH_PWROK signal goes inactive in S0 or DSW_PWROK drops	No	No	Yes	
SYS_PWROK Failure: SYS_PWROK signal goes inactive in S0	No	No	Yes	
Processor Thermal Trip (THERMTRIP#) causes transition to S5 and reset asserts	No	No	No	Yes
PCH internal thermal sensors signals a catastrophic temperature condition	No	No	No	Yes
Power Button 4 second override causes transition to S5 and reset asserts	No	No	No	Yes
Special shutdown cycle from processor causes CF9h-like PLTRST# and CF9h Global Reset Bit = 1	No	No	Yes	
Special shutdown cycle from processor causes CF9h-like PLTRST# and CF9h Global Reset Bit = 0 and CF9h (RST_CNT Register) Bit 3 = 1	No	Yes	No[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[4]	
Intel® Converged Security and Management Engine Triggered Host Reset without Power-Cycle	Yes	No	No[4]	
Intel® Converged Security and Management Engine Triggered Host Reset with Power-Cycle	No	Yes	No[4]	

continued...

Trigger	Host Reset Without Power Cycle[1]	Host Reset With Power Cycle[2]	Global Reset With Power Cycle[3]	Straight to S5[6] (Host Stays There)
Intel® Converged Security and Management Engine Triggered Power Button Override	No	No	No	Yes
Intel® Converged Security and Management Engine Watchdog Timer Timeout	No	No	No[8]	Yes
Intel® Converged Security and Management Engine Triggered Global Reset	No	No	Yes	
Intel® Converged Security and Management Engine Triggered Host Reset with power down (host stays there)	No	Yes[5]	No[4]	
PLTRST# Entry Timeout (Note 7)	No	No	Yes	
CPUPWRGD Stuck Low	No	No	Yes	
Power Management Watchdog Timer	No	No	No[8]	Yes
Intel® Converged Security and Management Engine Hardware Uncorrectable Error	No	No	No[8]	Yes

Notes:
1. The PCH drops this type of reset request if received while the system is in S4/S5.
2. PCH does not drop this type of reset request if received while system is in a software-entered 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 as per Deep Sx on page 122, the system will transition to Deep Sx.
7. PLTRST# Entry Timeout is automatically initiated if the hardware detects that the PLTRST# sequence has not been completed within 4 seconds of being started.
8. Trigger will result in Global Reset with Power-Cycle if AGR_LS_EN=1 and Global Reset occurred while the current or destination state was S0.

22.2 Signal Description

Signal Name	Type	Description
GPD1 / **ACPRESENT**	I	**ACPRESENT**: This input pin indicates when the platform is plugged into AC power or not. In addition to Intel® CSME to EC communication, the PCH uses this information to implement the Deep Sx policies. For example, the platform may be configured to enter Deep Sx when in S4 or S5 and only when running on battery. *Note:* An external pull-up resistor is required.
GPD0 / **BATLOW#**	I	**Battery Low**: An input from the battery to indicate that there is insufficient power to boot the system. Assertion will prevent wake from S4/S5 states or exit from Deep Sx state. This signal can also be enabled to cause an SMI# when asserted. This signal is multiplexed with GPD0. *Note:* For any platform not using this pin functionality, this signal must be tied high to VCCDSW_3P3. An external pull-up resistor to VCCDSW_3P3 is required.
GPP_B0 / **CORE_VID0**	O	**PCH Core VID Bit 0**: May connect to discrete VR on platform. In default mode this pin is driven high ('1').
GPP_B1 / **CORE_VID1**	O	**PCH Core VID Bit 1**: May connect to discrete VR on platform. In default mode this pin is driven high ('1').

continued...

Signal Name	Type	Description
GPP_H18 / **PROC_C10_GATE#**	O	**External Power Gate**: Control for VCCIO, VCCSTG and VCCPLL_OC during C10. When asserted, VCCIO, VCCSTG and VCCPLL_OC can be 0 V, however the power good indicators for these rails must remain asserted. *Note:* An external pull-down resistor is required.
DSW_PWROK	I	**DeepSx Well PWROK**: Power OK Indication for the VCCDSW_3p3 voltage rail. *Note:* This signal is in the RTC well. This signal cannot tie with RSMRST#.
GPD2 / **LAN_WAKE#**	I	**LAN WAKE**: An active low wake indicator from the Platform LAN Connect Device. *Note:* An external pull-up resistor is required.
GPD11 / **LANPHYPC**	O	**LAN PHY Power Control**: LANPHYPC is used to indicate that power needs to be restored to the Platform LAN Connect Device.
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. The platform may drive asynchronously. When PCH_PWROK is de-asserted, the PCH asserts PLTRST#. *Notes:* • PCH_PWROK must not glitch, even if RSMRST# is low • An external pull-down resistor is required.
GPP_B13 / **PLTRST#**	I	**Platform Reset**: The PCH asserts PLTRST# to reset devices on the platform. The PCH asserts PLTRST# low in Sx states and when a cold, warm, or global reset occurs. The PCH de-asserts PLTRST# upon exit from Sx states and the aforementioned resets. There is no guaranteed minimum assertion time for PLTRST#.
GPP_B11 / **PMCALERT#**	I/OD	**PMC Alert Pin**: Supports USB-C* PD controller architecture.
GPD3 / **PWRBTN#**	I	**Power Button**: The Power Button may cause an SMI# or SCI to indicate a system request to go to a sleep state. If the system is already in a sleep state, this signal will cause a wake event. If PWRBTN# is pressed for more than 4 seconds (default; timing is configurable), this will cause an unconditional transition (power button override) to the S5 state. Override will occur even if the system is in the S4 states. This signal has an internal Pull-up resistor and has an internal 16 ms de-bounce on the input. *Note:* Upon entry to S5 due to a power button override, if Deep Sx is enabled and conditions are met, the system will transition to Deep S5.
RSMRST#	I	**Primary Well Reset**: This signal is used for resetting the primary power plane logic. This signal must be asserted for at least 10 ms after the primary power wells are valid. When de-asserted, this signal is an indication that the primary power wells are stable. *Note:* An external pull down resistor is required
GPD6 **/ SLP_A#**	O	**SLP_A#**: Signal asserted when the Intel® CSME platform goes to M-Off or M3-PG. Depending on the platform, this pin may be used to control power to various devices that are part of the Intel® CSME sub-system in the platform. If you are not using SLP_A# for any functional purposes on your platform, or can tolerate lack of minimum assertion time, program the "SLP_A# minimum assertion width" value to the minimum. SLP_A# functionality can be utilized on the platform via either the physical pin or via the SLP_A# virtual wire over eSPI. *Note:* An external pull down resistor is required
SLP_LAN#	O	**LAN Sub-System Sleep Control**: When SLP_LAN# is de-asserted it indicates that the Platform LAN Connect Device must be powered. When SLP_LAN# is asserted, power can be shut off to the Platform LAN Connect Device. SLP_LAN# will always be de-asserted in S0 and anytime SLP_A# is de-asserted. *Note:* An external pull-down resistor is required.
GPD9 / **SLP_WLAN#**	O	**WLAN Sub-System Sleep Control**: When SLP_WLAN# is asserted, power can be shut off to the external wireless LAN device. SLP_WLAN# will always will be de-asserted in S0. If you are not using SLP_WLAN# for any functional purposes on your platform, or can tolerate lack of minimum assertion time, program the "SLP_A# minimum assertion width" value to the minimum.
GPP_B12 / **SLP_S0#**	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.

continued...

Signal Name	Type	Description
		Note: An external pull-up resistor is required.
GPD4 / **SLP_S3#**	O	**S3 Sleep Control:**:SLP_S3# is for power plane control. This signal shuts off power to all non-critical systems when in the S4, or S5 state. *Note:* An external pull-down resistor is required.
GPD5 / **SLP_S4#**	O	**S4 Sleep Control**: SLP_S4# is for power plane control. This signal shuts power to all non-critical systems when in the S4 or S5 state. *Notes:* • This pin must be used to control the DRAM power in order to use the PCH DRAM power-cycling feature. • An external pull-down resistor is required.
GPD10 / **SLP_S5#**	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 state. *Note:* An external pull-down resistor is required.
SLP_SUS#	O	**Deep Sx Indication**: When asserted (driven low), this signal indicates PCH is in Deep Sx state where internal primary power is shut off for enhanced power saving. When de-asserted (driven high), this signal indicates exit from Deep Sx state and primary power can be applied to PCH. For non- Deep Sx, this pin also needs to use to turn on VCCPRIM_1P8 VR. This pin cannot left unconnected. *Notes:* • This is in the DSW power well • An external pull-down resistor is required.
SPIVCCIOSEL	I	**SPI Operation Voltage Select** There is no internal pull-up or pull-down on the strap. An external resistor is required. 0 = SPI voltage is 3.3 V (4.7 kohm pull-down to GND), 1 = SPI voltage is 1.8V (4.7 kohm pull-up to VCCDSW_3p3).
GPP_A3 / ESPI_IO3 / **SUSACK#**	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.
GPD8 / **SUSCLK**	O	**Suspend Clock**: This clock is a digitally buffered version of the RTC clock.
GPP_A2 / ESPI_IO2 / **SUSWARN#** / SUSPWRDNACK	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 primary well power loss on a falling edge and preparing for Primary well related activity (host/Intel CSME wakes and runtime events) on a rising edge. SUSACK# must be driven to match SUSWARN# once the above preparation is complete. SUSACK# should be asserted within a minimal amount of time from SUSWARN# assertion as no wake events are supported if SUSWARN# is asserted but SUSACK# is not asserted. Platforms supporting Deep Sx, but not wishing to participate in the handshake during wake and Deep Sx entry may tie SUSACK# to SUSWARN#. This pin is multiplexed with SUSPWRDNACK since it is not needed in Deep Sx supported platforms.
GPP_A2 / ESPI_IO2 / SUSWARN# / **SUSPWRDNACK**	O	**SUSPWRDNACK**: Active high. Asserted by the PCH on behalf of the Intel CSME when it does not require the PCH Primary well to be powered. Platforms are not expected to use this signal when the PCH Deep Sx feature is used.
GPP_H3 / **SX_EXIT_HOLDOFF#**	I	**Sx Exit Holdoff Delay**: Delay exit from Sx state after SLP_A# is de-asserted. *Note:* When eSPI is enabled, the flash sharing functionality using SX_EXIT_HOLDOFF# is not supported, but the pin still functions to hold off Sx exit after SLP_A# de-assertion.
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. *Note:* An external pull-down resistor is required
SYS_RESET#	I	**System Reset**: This pin forces an internal reset after being de-bounced.

continued...

Signal Name	Type	Description
		Note: An external pull-up resistor is required.
GPP_B15 / **TIME_SYNC0**/ ISH_GP7	I	**Time Synchronization**: Used for synchronization both input (latch time when pin asserted) and output (toggle pin when programmed time is hit).
GPP_B14/ **TIME_SYNC1** / SPKR / SATA_LED# / ISH_GP6	I	**Time Synchronization**: Used for synchronization both input (latch time when pin asserted) and output (toggle pin when programmed time is hit).
GPP_B2 / **VRALERT#**	I	**VR Alert**: ICC Max. throttling indicator from the PCH voltage regulators. VRALERT# pin allows the VR to force PCH throttling to prevent an over current shutdown. PMC based on the VRALERT# and messages from the processor. The messages from the processor allows the processor to constrain the PCH to a particular power budget.
WAKE#	I/OD	**PCI Express* Wake Event in Sx**: Input Pin in Sx. Sideband wake signal on PCI Express* asserted by components requesting wake up. *Notes:* • This is an output pin during S0ix states hence this pin can not be used to wake up the system during S0ix states. • An external pull-up resistor is required.
VCCST_OVERRIDE	O	**VccST Override**: Signal that allows the PCH to keep VCCST powered ON (in case VCCST is powered down) for USB-C wake capability (connected to VCCSTPWRGOOD_TCSS on board). Signal will stay high when plug-in device on USB Type-C Subsystem port and signal will stay low when no device is connected.
GPP_F22 / **VNN_CTRL**		**VNN_Control**: External bypass rail control pin. Without requiring BIOS to be involved during the S0ix states. This pin use to control of the VCC_VNNEXT_1P05 voltage.
GPP_F23 / **V1p05_CTRL**		**V1p05_Control**: External bypass rail control pin. Without requiring BIOS to be involved during the S0ix states. This pin use to control of the VCC_V1P05EXT_1P05 voltage.

22.3 Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value	Notes
ACPRESENT	Pull-down	15 kohm - 40 kohm	1
LAN_WAKE#	Pull-down	15 kohm - 40 kohm	1
PWRBTN#	Pull-up	20 kohm +/- 30%	
SUSACK#	Pull-up	20 kohm+/- 30%	
WAKE#	Pull-down	15 kohm - 40 kohm	1
Note: 1. Pull-down is configurable and can be enabled in Deep Sx state; refer to DSX_CFG register for more details.			

22.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[18]	Immediately after Reset[18]	S4/S5	Deep Sx
ACPRESENT[6,10,15]	DSW	Undriven /Driven Low4	Undriven	Undriven	Undriven/Internal Pull-down[8]
BATLOW#	DSW	Undriven	Undriven	Undriven	OFF
CORE_VID0[11]	Primary	Driven High	Driven High	Driven High	OFF
CORE_VID1[11]	Primary	Driven High	Driven High	Driven High	OFF
PROC_C10 [1]	Primary	Driven High	Driven High	Driven Low	OFF

continued...

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

continued...

Signal Name	Power Plane	During Reset[18]	Immediately after Reset[18]	S4/S5	Deep Sx
VRALERT# [15]	Primary	Undriven	Undriven	Undriven	OFF
WAKE# [13]	DSW	Undriven	Undriven	Undriven	Undriven/Internal Pull-down

Notes:
1. Driven High during S0 and driven Low during S0i3 when all criteria for assertion are met.
2. SLP_S4# is driven low in S4/S5.
3. SLP_S5# is driven high in S4, driven low in S5.
4. In non-Deep Sx mode, pin is driven low.
5. Based on wake events and Intel® CSME state. SUSPWRDNACK is always '0' while in M0 or M3, but can be driven to '0' or '1' while in Moff state. SUSPWRDNACK is the default mode of operation. If Deep Sx is supported, then subsequent boots will default to SUSWARN#.
6. The pin requires glitch-free output sequence. The pad should only be pulled low momentarily when the corresponding buffer power supply is not stable.
7. Based on wake event and Intel CSME state.
8. Pull-down is configurable and can be enabled in Deep Sx state; refer to DSX_CFG register for more details.
9. When platform enters Deep Sx, the SLP_S4# and SLP_S5# pin will retain the value it held prior to Deep Sx entry.
10. Internal weak pull-down resistor is enabled during power sequencing.
11. The CORE_VID pins defaults to '1' and will be driven to '1' to reflect that voltage will support 1.8 V. The VID able to change to 1.8 V/ 3.3 V based on the CPU and the state.
12. Pin state is a function of whether the platform is configured to have Intel CSME on or off in Sx.
13. Output High-Z, not glitch free.
14. Output High-Z, glitch free with ~1 k Pull-down during respective power sequencing
15. Output High-Z, not glitch free.
16. Output High-Z, glitch free with ~20 k Pull-down during respective power sequencing.
17. Reset reference for primary well pins is RSMRST#, DSW well pins is DSW_PWROK, and RTC well pins is RTCRST#.
18. Sx can be optionally be high when RSMRST# is high and the buffer moves to its native mode at which point it will become low.

23.0 Real Time Clock (RTC)

The PCH contains a real-time clock functionally compatible with the Motorola* MC146818B. The real-time clock has 256 bytes of battery-backed RAM. The real-time clock performs two key functions

- keep track of the time of day
- store system data even when the system is powered down as long as the RTC power well is powered

The RTC operates on a 32.768 kHz oscillating source and a 3 V battery or system battery if configured by design as the source.

The RTC also supports two lockable memory ranges. By setting bits in the configuration space, two 8-byte ranges can be locked to read and write accesses. This prevents unauthorized reading of passwords or other system security information.

The RTC also supports a date alarm that allows for scheduling a wake up event up to month in advance.

Table 63. Acronyms

Acronyms	Description
BCD	Binary Coded Decimal
CMOS	Complementary Metal Oxide Semiconductor. A manufacturing process used to produce electronics circuits, but in reference to RTC is used interchangeably as the RTC's RAM i.e. clearing CMOS meaning to clear RTC RAM.
ESR	Equivalent Series Resistance. Resistive element in a circuit such as a clock crystal.
GPI	General Purpose Input
PPM	Parts Per Million. Used to provide crystal accuracy or as a frequency variation indicator.
RAM	Random Access Memory

23.1 Signal Description

Signal Name	Type	Description
RTCX1	I	**Crystal Input 1:** This signal is connected to the 32.768 kHz crystal (max 50K Ohm ESR). If no external crystal is used, then RTCX1 can be driven with the desired clock rate. Maximum voltage allowed on this pin is 1.5 V.
RTCX2	O	**Crystal Input 2:** This signal is connected to the 32.768 kHz crystal (max 50K Ohm ESR). If no external crystal is used, then RTCX2 must be left floating.
RTCRST#	I	**RTC Reset**: When asserted, this signal resets register bits in the RTC well.

continued...

Signal Name	Type	Description
		Notes: 1. Unless CMOS is being cleared (only to be done in the G3 power state) with a jumper, the RTCRST# input must always be high when all other RTC power planes are on. 2. In the case where the RTC battery is dead or missing on the platform, the RTCRST# pin must rise before the DSW_PWROK pin.
SRTCRST#	I	**Secondary RTC Reset**: This signal resets the manageability register bits in the RTC well when the RTC battery is removed. *Notes:* 1. The SRTCRST# input must always be high when all other RTC power planes are on. 2. In the case where the RTC battery is dead or missing on the platform, the SRTCRST# pin must rise before the DSW_PWROK pin. 3. SRTCRST# and RTCRST# should not be shorted together.

23.2 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
RTCRST#	RTC	Undriven	Undriven	Undriven	Undriven
SRTCRST#	RTC	Undriven	Undriven	Undriven	Undriven
Note: 1. Reset reference for RTC well pins is RTCRST#.					

24.0 Serial ATA (SATA)

The PCH SATA controller support two modes of operation, AHCI mode using memory space and RAID mode. The PCH SATA controller does not 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.

Table 64. 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

Table 65. 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

24.1 Functional Description

The PCH SATA host controller (D23:F0) supports AHCI or RAID mode.

The PCH SATA controller does not support legacy IDE mode or combination mode.

The PCH SATA controller interacts with an attached mass storage device through a register interface that is compatible with an SATA AHCI/RAID host adapter. The host software follows existing standards and conventions when accessing the register interface and follows standard command protocol conventions.

24.1.1 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.

24.1.2 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)

24.1.3 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 the power savings in the low power states while enabling Hot - Plug operation. Refer to Chapter 7 of the AHCI specification for details.

24.1.4 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 - P I/O Capabilities table and PCH - P SKUs table for details on feature availability.

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.
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.

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.

24.1.5 Power Management Operation

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

Power State Mappings

The D0 PCI Power Management (PM) state for device is supported by the PCH SATA controller.

SATA devices may also have multiple power states. SATA adopted 3 main power states from parallel ATA. The three device states are supported through ACPI. They are:

- **D0** – Device is working and instantly available.

- **D1** – Device enters when it receives a STANDBY IMMEDIATE command. Exit latency from this state is in seconds.
- **D3** – From the SATA device's perspective, no different than a D1 state, in that it is entered using the STANDBY IMMEDIATE command. However, an ACPI method is also called which will reset the device and then cut its power.

Each of these device states are subsets of the host controller's D0 state.

Finally, the SATA specification defines three PHY layer power states, which have no equivalent mappings to parallel ATA. They are:

- **PHY READY** – PHY logic and PLL are both on and in active state.
- **Partial** – PHY logic is powered up, and in a reduced power state. The link PM exit latency to active state maximum is 10 ns.
- **Slumber** – PHY logic is powered up, and in a reduced power state. The link PM exit latency to active state maximum is 10 ms.
- **Devslp** – PHY logic is powered down. The link PM exit latency from this state to active state maximum is 20 ms, unless otherwise specified by DETO in Identify Device Data Log page 08h (Refer to 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.

Power State Transitions

- **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.

- **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.

- **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 is most likely to be used in ATA/ATAPI is the "STANDBY IMMEDIATE" command.

- **Host Controller $D3_{HOT}$ State**

  After the interface and device have been put into a low power state, the SATA host controller may be put into a low power state. This is performed using the PCI power management registers in configuration space. There are two very important aspects to Note when using PCI power management:

  1. When the power state is D3, only accesses to configuration space are allowed. Any attempt to access the memory or I/O spaces will result in master abort.
  2. When the power state is D3, no interrupts may be generated, even if they are enabled. If an interrupt status bit is pending when the controller transitions to D0, an interrupt may be generated.

  When the controller is put into D3, it is assumed that software has properly shut down the device and disabled the ports. Therefore, there is no need to sustain any values on the port wires. The interface will be treated as if no device is present on the cable, and power will be minimized.

  When returning from a D3 state, an internal reset will not be performed.

Low Power Platform Consideration

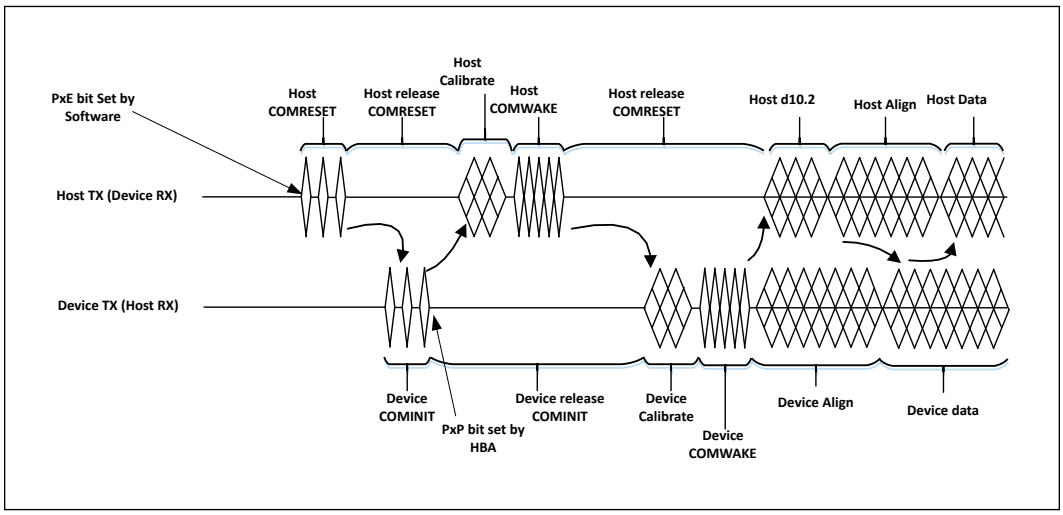
When low power feature is enabled, the Intel SATA controller may power off PLLs or OOB detection circuitry while in the Slumber link power state. As a result, a device initiated wake may not be recognized by the host. For example, when the low power feature is enabled it can prevent a Zero Power ODD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

24.1.6 SATA Device Presence

The flow used to indicate SATA device presence is shown in the Figure below. 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 15. Flow for Port Enable/Device Present Bits



24.1.7 SATA LED

The SATALED# output is driven whenever the BSY bit is set in any SATA port. The SATALED# is an active - low open - drain output. When SATALED# is low, the LED should be active. When SATALED# is high, the LED should be inactive.

24.1.8 Advanced Host Controller Interface (AHCI) Operation

The PCH SATA controller provides hardware support for Advanced Host Controller Interface (AHCI), a standardized programming interface for SATA host controllers developed through a joint industry effort. Platforms supporting AHCI may take advantage of performance features such as port independent DMA Engines—each device is treated as a master—and hardware-assisted native command queuing.

AHCI defines transactions between the SATA controller and software and enables advanced performance and usability with SATA. Platforms supporting AHCI may take advantage of performance features such as no master/slave designation for SATA devices—each device is treated as a master—and hardware assisted native command queuing. AHCI also provides usability enhancements such as hot - plug and advanced power management. AHCI requires appropriate software support (such as, an AHCI driver) and for some features, hardware support in the SATA device or additional platform hardware. Visit the Intel web site for current information on the AHCI specification.

The PCH SATA controller supports all of the mandatory features of the *Serial ATA Advanced Host Controller Interface Specification*, Revision 1.3.1 and many optional features, such as hardware assisted native command queuing, aggressive power management, LED indicator support, and hot - plug through the use of interlock switch support (additional platform hardware and software may be required depending upon the implementation).

NOTE

For reliable device removal notification while in AHCI operation without the use of interlock switches (surprise removal), interface power management should be disabled for the associated port. Refer to Section 7.3.1 of the AHCI Specification for more information.

24.1.9 Enclosure Management (SGPIO Signals)

Enclosure management is a mechanism by which the storage driver can monitor and control auxiliary service in a drive enclosure. This feature is only valid in AHCI/RAID mode.

The SGPIO signals are used in the enclosure management protocol (refer to the SFF-8485 specification) and supports multiple - activity LEDs to shows the per drive status information.

NOTE

These signals are not related to SATALED#.

The SGPIO group interfaces with an external controller chip that fetches and serializes the data for driving across the SGPIO bus. The output signals then control the LEDs within the enclosure. The PCH SATA controller only supports LED messages transmission and has three SGPIO protocol signals implemented, that is SCLOCK, SDATAOUT and SLOAD.

NOTE

Intel does not validate all possible usage cases of this feature. Customers should validate their specific design implementation on their own platforms.

Mechanism

The enclosure management for SATA Controller involves sending messages that control LEDs in the enclosure. The messages for this function are stored after the normal registers in the AHCI BAR, at Offset 580h bytes for the PCH from the beginning of the AHCI BAR as specified by the EM_LOC global register.

Software creates messages for transmission in the enclosure management message buffer. The data in the message buffer should not be changed if CTL.TM bit is set by software to transmit an update message. Software should only update the message buffer when CTL.TM bit is cleared by hardware otherwise the message transmitted will be indeterminate. Software then writes a register to cause hardware to transmit the message or take appropriate action based on the message content. The software should only create message types supported by the controller, which is LED messages for the PCH. If the software creates other non LED message types (such as, SAF-TE, SES-2), the SGPIO interface may hang and the result is indeterminate.

During reset all SGPIO pins will be in tri - state state. The interface will continue staying in tri - state state after reset until the first transmission occurs, when software programs the message buffer and sets the transmit bit CTL.TM. The SATA host controller will initiate the transmission by driving SCLOCK and at the same time driving the SLOAD to **0** prior to the actual bit stream transmission. The Host will drive

SLOAD low for at least 5 SCLOCK then only start the bit stream by driving the SLOAD to high. SLOAD will be driven high for 1 SCLOCK, followed by vendor-specific pattern that is default to "0000" if software is yet to program the value. A total of 24-bit streams from 8 ports (Port 0, Port 1, Port 2, Port 3, Port 4, Port 5, Port 6, Port 7) of 3-bit per port LED message will be transmitted on SDATAOUT0 pin after the SLOAD is driven high for 1 SCLOCK. For 8 SATA port configuration, only 4 ports (port 4, port 5, port 6 and port 7) of 12 bit total LED message follow by 12 bits of tri-state value will be transmitted out on SDATAOUT1 pin. For 6 SATA port configuration, only 2 ports (port 4 and port 5) of 6 bit total LED message follow by 18 bits of tri-state value will be transmitted out on SDATAOUT1 pin. For 4 SATA port configuration, SDATAOUT1 pin is not required hence can be tri-state always.

All the default LED message values will be high prior to software setting them, except the Activity LED message that is configured to be hardware driven that will be generated based on the activity from the respective port. All the LED message values will be driven to '1' for the port that is unimplemented as indicated in the Port Implemented register regardless of the software programmed value through the message buffer.

There are 2 different ways of resetting the PCH's SGPIO interface, asynchronous reset and synchronous reset. Asynchronous reset is caused by platform reset to cause the SGPIO interface to be tri-state asynchronously. Synchronous reset is caused by setting the CTL.RESET bit, or HBA reset, where Host Controller will complete the existing full bit stream transmission then only tri-state all the SGPIO pins. After the reset, both synchronous reset and asynchronous reset, the SGPIO pins will stay tri-stated.

NOTE

The PCH Host Controller does not ensure that it will cause the target SGPIO device or controller to be reset. Software is responsible to keep the PCH SGPIO interface in tri-state for 2 second to cause a reset on the target of the SGPIO interface.

Message Format

Messages shall be constructed with a one DWord header that describes the message to be sent followed by the actual message contents. The first DWord shall be constructed as shown in Enclosure Management Message Format (EM_MF) register

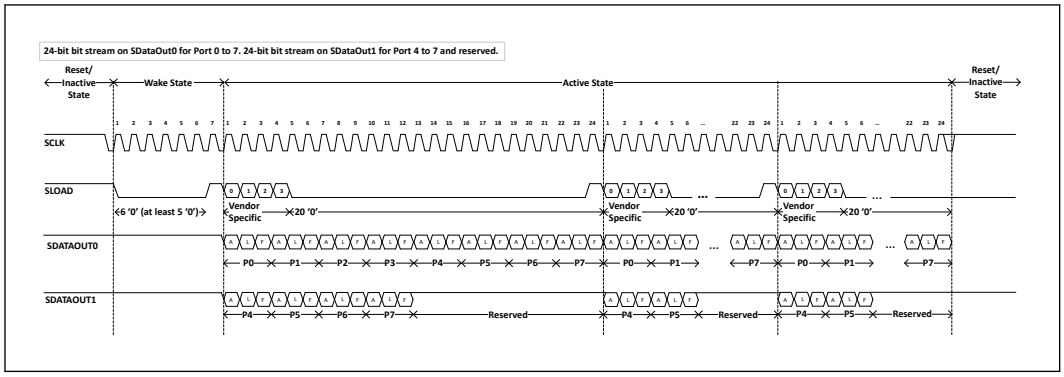
The SAF-TE, SES-2, and SGPIO message formats are defined in the corresponding specifications, respectively. The LED message type is defined in the Enclosure Management LED (EM_LED) registe. It is the responsibility of software to ensure the content of the message format is correct. If the message type is not programmed as 'LED' for this controller, the controller shall not take any action to update its LEDs. For LED message type, the message size always consists of 4 bytes.

LED Message Type

The LED message type specifies the status of up to three LEDs. Typically, the usage for these LEDs is activity, fault, and locate. Not all implementations necessarily contain all LEDs (for example, some implementations may not have a locate LED). The message identifies the HBA port number and the Port Multiplier port number that the slot status applies to. If a Port Multiplier is not in use with a particular device, the Port Multiplier port number shall be '0'. The format of the LED message type is defined in the Enclosure Management LED (EM_LED) registe. The LEDs shall retain their values until there is a following update for that particular slot.

SGPIO Waveform

Figure 16. Serial Data Transmitted over SGPIO Interface



24.2 Signals Description

Signal Name	Type	Description
GPP_E4 / **SATA_DEVSLP0**	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. *Note:* This pin can be mapped to SATA Port 0.
GPP_E5 / **SATA_DEVSLP1**	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. *Note:* This pin can be mapped to SATA Port 1.
PCIE11_TXN / **SATA0_TXN** PCIE11_TXP / **SATA0_TXP**	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.
PCIE11_RXN / **SATA0_RXN** PCIE11_RXP / **SATA0_RXP**	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.
PCIE12_TXN / **SATA1_TXN** PCIE12_TXP / **SATA1_TXP**	O	**Serial ATA Differential Transmit Pair 1** :These outbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE12_RXN / **SATA1_RXN** PCIE12_RXP / **SATA1_RXP**	I	**Serial ATA Differential Receive Pair 1**: These inbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
GPP_E0 / SATAXPCIE0 / **SATAGP0**	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.
GPP_A12 / SATAXPCIE1 / **SATAGP1**	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.

continued...

Signal Name	Type	Description
		Note: This default use of this pin is GPP_A12. Pin defaults to Native mode as SATAXPCIE1 depends on soft-strap.
GPP_B14 / SPKR / TIME_SYNC1 / **SATA_LED#** / ISH_GP6	OD	**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.

24.3 Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type
SATAXPCIE[0:1]	Internal pull-up
Note: Internal Pull-Up Resistors are 20 kohm ± 30% unless specified.	

24.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[3]	Immediately after Reset[3]	S4/S5	Deep Sx
SATA[0:1]_TXN SATA[0:1]_TXP SATA[0:1]_RXN SATA[0:1]_RXP	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SATA_LED#	Primary	Undriven	Undriven	Undriven	OFF
SATA_DEVSLP[0:1][1]	Primary	Undriven	Undriven	Driven Low	OFF
SATAGP[0:1][2]	Primary	Undriven	Undriven	Undriven	OFF
SATAXPCIE[0:1][2]	Primary	Internal Pull-up	Internal Pull-up	Undriven	OFF

Notes:
1. Pin defaults to GPIO mode. The pin state during and immediately after reset follows default GPIO mode pin state. The pin state for S0 to Deep Sx reflects assumption that GPIO Use Select register was programmed to native mode functionality. If GPIO Use Select register is programmed to GPIO mode, refer to Multiplexed GPIO (Defaults to GPIO Mode) section for the respective pin states in S0 to Deep Sx.
2. Pin defaults to Native mode as SATAXPCIEx depends on soft-strap.
3. Reset reference for primary well pins is RSMRST#.

25.0 System Management Interface and SMLink

The PCH provides two SMLink interfaces, SMLink0 and SMLink1. The interfaces are intended for system management and are controlled by the Intel® CSME. Refer to System Management on page 27 for more detail.

Table 66. Acronyms

Acronyms	Description
BMC	Baseboard Management Controller
EC	Embedded Controller

25.1 Functional Description

The SMLink interfaces are controlled by the Intel® CSME.

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

SMLink1 can be used with an Embedded Controller (EC) or Baseboard Management Controller (BMC).

Both SMLink0 and SMLink1 support up to 1 MHz.

NOTE

Access to the PCH thermal sensor should be via eSPI. as the SMLink 1 is disabled in Consumer platforms.

25.1.1 Integrated USB-C Usage

SMLink1 is used to communicate with USB-C* PD Controller on the platform to configure different modes such as USB, Display Port, Thunderbolt*, etc. When used for Integrated USB-C purposes, a soft strap must be set to indicate that integrated USB-C ports from CPU are being used.

SMLINK1 uses master mode and gets an alert signal from PMCALERT#.

Based on capabilities of different PD Controllers, re-timers needed for USB-C connector on the platform may need to be controlled by SoC also. In these cases, both PD Controller and Re-timers will be connected to SMLink1. SMLink1 is used for all USB-C connectors on the platform.

USB-C connectors are present on edges of systems and could also be on opposite ends, so (SMLink1, PMCAlert) could be routed to long distance on the motherboard provided total bus capacitance specification is met.

USB-C Re-timer control (like Firmware Load, USB-C configuration) handling depends on the number of I^2C ports available on the PD controller.

If the PD controller has two I^2C ports then PCH PMC will handle the Re-timer and PD controller, but if the PD controller has three or more I^2C ports then PCH PMC will handle only PD controller. Re-timers can be handled by PD controller.

SMLink1 should be run at 400 kHz when used for USB-C purposes.

25.2 Signal Description

Signal Name	Type	Description
INTRUDER#	I	**Intruder Detect**: This signal can be set to disable the system if box detected open.
GPP_C4 / **SML0DATA**	I/OD	**System Management Link 0 Data**: SMBus link to external PHY. External Pull-up resistor required.
GPP_C3/**SML0CLK**	I/OD	**System Management Link 0 Clock** External Pull-up resistor required.
GPP_C5 / **SML0ALERT#**	I/OD	**System Management 0 Alert**: Alert for the SMBus controller to optional Embedded Controller or BMC. External Pull-up resistor required.
GPP_C6 / **SML1CLK**	I/OD	**System Management Link 1 Clock**: SMBus link to optional Embedded Controller or BMC. External Pull-up resistor required.
GPP_C7 / **SML1DATA**	I/OD	**System Management Link 1 Data**: SMBus link to optional Embedded Controller or BMC. External Pull-up resistor required.
GPP_B23 / **SML1ALERT#** / PCHHOT#	I/OD	**System Management 1 Alert**: Alert for the SMBus controller to optional Embedded Controller or BMC. A soft-strap determines the native function SML1ALERT# or PCHHOT# usage. This is **NOT** the right Alert pin for USB-C* usage. External Pull-up resistor is required on this pin.
GPP_B11/**PMCALERT#**	I/OD	**USB Type-C* PD Controller / Re-timer Alert**: Alert for the SMLink1 Bus controller to all USB Type-C* PD Controllers, mandatory requirement for integrated USB-C* feature to work. External Pull-up resistor is required on this pin.

25.3 Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value	Notes
SML[1:0]ALERT#	Pull-down	20 kohm ± 30%	The internal pull-down resistor is disable after RSMRST# de-asserted.
PCHHOT#	Pull-down	20 kohm ± 30%	The internal pull-down resistor is disable after RSMRST# de-asserted.

25.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	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	Pull-down (Internal)	Driven Low	Pull-down (Internal)	OFF
PCHHOT#	Primary	Pull-down (Internal)	Driven Low	Pull-down (Internal)	OFF
PMCALERT#	Primary	Undriven	Undriven	Undriven	OFF
Note: 1. Reset reference for primary well pins is RSMRST# and RTC well pins is RTCRST#.					

26.0 Host System Management Bus (SMBus) Controller

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

The host SMBus controller supports up to 100 kHz clock speed.

Table 67. Acronyms

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

Table 68. References

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

26.1 Functional Description

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

- **Host Controller:** Provides a mechanism for the processor to initiate communications with SMBus peripherals (slaves). The PCH is also capable of operating in a mode in which it can communicate with I^2C compatible devices.
- **Slave Interface:** Allows an external master to read from or write to the PCH. Write cycles can be used to cause certain events or pass messages, and the read cycles can be used to determine the state of various status bits. The PCH's internal host controller cannot access the PCH's internal Slave Interface.

26.1.1 Host Controller

The host SMBus controller supports up to 100 - KHz clock speed and is clocked by the RTC clock.

The PCH can perform SMBus messages with either Packet Error Checking (PEC) enabled or disabled. The actual PEC calculation and checking is performed in SW. The SMBus host controller logic can automatically append the CRC byte if configured to do so.

The SMBus Address Resolution Protocol (ARP) is supported by using the existing host controller commands through software, except for the Host Notify command (which is actually a received message).

The PCH SMBus host controller checks for parity errors as a target. If an error is detected, the detected parity error bit in the PCI Status Register is set.

Host Controller Operation Overview

The SMBus host controller is used to send commands to other SMBus slave devices. Software sets up the host controller with an address, command, and, for writes, data and optional PEC; and then tells the controller to start. When the controller has finished transmitting data on writes, or receiving data on reads, it generates an SMI# or interrupt, if enabled.

The host controller supports eight command protocols of the SMBus interface (refer to the System Management Bus (SMBus) Specification, Version 2.0): Quick Command, Send Byte, Receive Byte, Write Byte/Word, Read Byte/Word, Process Call, Block Read/Write, and Block Write–Block Read Process Call.

The SMBus host controller requires that the various data and command fields be setup for the type of command to be sent. When software sets the START bit, the SMBus Host controller performs the requested transaction, and interrupts the processor (or generates an SMI#) when the transaction is completed. Once a START command has been issued, the values of the "active registers" (Host Control, Host Command, Transmit Slave Address, Data 0, Data 1) should not be changed or read until the interrupt status message (INTR) has been set (indicating the completion of the command). Any register values needed for computation purposes should be saved prior to issuing of a new command, as the SMBus host controller updates all registers while completing the new command.

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

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

Command Protocols

In all of the following commands, the Host Status Register (offset 00h) is used to determine the progress of the command. While the command is in operation, the HOST_BUSY bit is set. If the command completes successfully, the INTR bit will be set in the Host Status Register. If the device does not respond with an acknowledge, and the transaction times out, the DEV_ERR bit is set.

If software sets the KILL bit in the Host Control Register while the command is running, the transaction will stop and the FAILED bit will be set after the PCH forces a time - out. In addition, if KILL bit is set during the CRC cycle, both the CRCE and DEV_ERR bits will also be set.

Quick Command

When programmed for a Quick Command, the Transmit Slave Address Register is sent. The PEC byte is never appended to the Quick Protocol. Software should force the PEC_EN bit to 0 when performing the Quick Command. Software must force the I2C_EN bit to 0 when running this command. Refer to Section 5.5.1 of the System Management Bus (SMBus) Specification, Version 2.0 for the format of the protocol.

Send Byte/Receive Byte

For the Send Byte command, the Transmit Slave Address and Device Command Registers are sent. For the Receive Byte command, the Transmit Slave Address Register is sent. The data received is stored in the DATA0 register. Software must force the I2C_EN bit to 0 when running this command.

The Receive Byte is similar to a Send Byte, the only difference is the direction of data transfer. Refer to Sections 5.5.2 and 5.5.3 of the System Management Bus *(SMBus)* Specification, Version 2.0 for the format of the protocol.

Write Byte/Word

The first byte of a Write Byte/Word access is the command code. The next 1 or 2 bytes are the data to be written. When programmed for a Write Byte/Word command, the Transmit Slave Address, Device Command, and Data0 Registers are sent. In addition, the Data1 Register is sent on a Write Word command. Software must force the I2C_EN bit to 0 when running this command. Refer to Section 5.5.4 of the *System Management Bus (SMBus) Specification*, Version 2.0 for the format of the protocol.

Read Byte/Word

Reading data is slightly more complicated than writing data. First the PCH must write a command to the slave device. Then it must follow that command with a repeated start condition to denote a read from that device's address. The slave then returns 1 or 2 bytes of data. Software must force the I2C_EN bit to 0 when running this command.

When programmed for the read byte/word command, the Transmit Slave Address and Device Command Registers are sent. Data is received into the DATA0 on the read byte, and the DAT0 and DATA1 registers on the read word. Refer to Section 5.5.5 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

Process Call

The process call is so named because a command sends data and waits for the slave to return a value dependent on that data. The protocol is simply a Write Word followed by a Read Word, but without a second command or stop condition.

When programmed for the Process Call command, the PCH transmits the Transmit Slave Address, Host Command, DATA0 and DATA1 registers. Data received from the device is stored in the DATA0 and DATA1 registers.

The Process Call command with I2C_EN set and the PEC_EN bit set produces undefined results. Software must force either I2C_EN or PEC_EN to 0 when running this command. Refer to Section 5.5.6 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

NOTES

1. For process call command, the value written into bit 0 of the Transmit Slave Address Register needs to be 0.
2. If the I2C_EN bit is set, the protocol sequence changes slightly, the Command Code (Bits 18:11 in the bit sequence) are not sent. As a result, the slave will not acknowledge (Bit 19 in the sequence).

Block Read/Write

The PCH contains a 32 - byte buffer for read and write data which can be enabled by setting bit 1 of the Auxiliary Control register at offset 0Dh in I/O space, as opposed to a single byte of buffering. This 32 - byte buffer is filled with write data before transmission, and filled with read data on reception. In the PCH, the interrupt is generated only after a transmission or reception of 32 bytes, or when the entire byte count has been transmitted/received.

The byte count field is transmitted but ignored by the PCH as software will end the transfer after all bytes it cares about have been sent or received.

For a Block Write, software must either force the I2C_EN bit or both the PEC_EN and AAC bits to 0 when running this command.

The block write begins with a slave address and a write condition. After the command code the PCH issues a byte count describing how many more bytes will follow in the message. If a slave had 20 bytes to send, the first byte would be the number 20 (14h), followed by 20 bytes of data. The byte count may not be 0. A Block Read or Write is allowed to transfer a maximum of 32 data bytes.

When programmed for a block write command, the Transmit Slave Address, Device Command, and Data0 (count) registers are sent. Data is then sent from the Block Data Byte register; the total data sent being the value stored in the Data0 Register.

On block read commands, the first byte received is stored in the Data0 register, and the remaining bytes are stored in the Block Data Byte register. Refer to section 5.5.7 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

NOTE

For Block Write, if the I2C_EN bit is set, the format of the command changes slightly. The PCH will still send the number of bytes (on writes) or receive the number of bytes (on reads) indicated in the DATA0 register. However, it will not send the contents of the DATA0 register as part of the message. When operating in I^2C mode (I2C_EN bit is set), the PCH will never use the 32 - byte buffer for any block commands.

I^2C* Read

This command allows the PCH to perform block reads to certain I^2C devices, such as serial E^2PROMs. The SMBus Block Read supports the 7 - bit addressing mode only.

However, this does not allow access to devices using the I^2C “Combined Format” that has data bytes after the address. Typically these data bytes correspond to an offset (address) within the serial memory chips.

NOTE

This command is supported independent of the setting of the I2C_EN bit. The I^2C Read command with the PEC_EN bit set produces undefined results. Software must force both the PEC_EN and AAC bit to 0 when running this command.

For I^2C Read command, the value written into bit 0 of the Transmit Slave Address Register (SMB I/O register, offset 04h) needs to be 0.

The format that is used for the command is shown in the table below:

Table 69. I²C* Block Read

Bit	Description
1	Start
8:2	Slave Address – 7 bits
9	Write
10	Acknowledge from slave
18:11	Send DATA1 register
19	Acknowledge from slave
20	Repeated Start
27:21	Slave Address – 7 bits
28	Read
29	Acknowledge from slave
37:30	Data byte 1 from slave – 8 bits
38	Acknowledge
46:39	Data byte 2 from slave – 8 bits
47	Acknowledge
–	Data bytes from slave/Acknowledge
–	Data byte N from slave – 8 bits
–	NOT Acknowledge
–	Stop

The PCH will continue reading data from the peripheral until the NAK is received.

Block Write – Block Read Process Call

The block write - block read process call is a two - part message. The call begins with a slave address and a write condition. After the command code the host issues a write byte count (M) that describes how many more bytes will be written in the first part of the message. If a master has 6 bytes to send, the byte count field will have the value 6 (0000 0110b), followed by the 6 bytes of data. The write byte count (M) cannot be 0.

The second part of the message is a block of read data beginning with a repeated start condition followed by the slave address and a Read bit. The next byte is the read byte count (N), which may differ from the write byte count (M). The read byte count (N) cannot be 0.

The combined data payload must not exceed 32 bytes. The byte length restrictions of this process call are summarized as follows:

- M ≥ 1 byte
- N ≥ 1 byte
- M + N ≤ 32 bytes

The read byte count does not include the PEC byte. The PEC is computed on the total message beginning with the first slave address and using the normal PEC computational rules. It is highly recommended that a PEC byte be used with the Block Write - Block Read Process Call. Software must do a read to the command register (offset 2h) to reset the 32 byte buffer pointer prior to reading the block data register.

NOTES

1. There is no STOP condition before the repeated START condition, and that a NACK signifies the end of the read transfer.
2. E32B bit in the Auxiliary Control register must be set when using this protocol.

Refer to Section 5.5.8 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

Bus Arbitration

Several masters may attempt to get on the bus at the same time by driving the SMBDATA line low to signal a start condition. The PCH continuously monitors the SMBDATA line. When the PCH is attempting to drive the bus to a 1 by letting go of the SMBDATA line, and it samples SMBDATA low, then some other master is driving the bus and the PCH will stop transferring data.

If the PCH detects 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, generates an interrupt or SMI#. The processor is responsible for restarting the transaction.

Clock Stretching

Some devices may not be able to handle their clock toggling at the rate that the PCH as an SMBus master would like. They have the capability of stretching the low time of the clock. When the PCH attempts to release the clock (allowing the clock to go high), the clock will remain low for an extended period of time.

The PCH monitors the SMBus clock line after it releases the bus to determine whether to enable the counter for the high time of the clock. While the bus is still low, the high time counter must not be enabled. Similarly, the low period of the clock can be stretched by an SMBus master if it is not ready to send or receive data.

Bus Timeout (PCH as SMBus Master)

If there is an error in the transaction, such that an SMBus device does not signal an acknowledge or holds the clock lower than the allowed Timeout time, the transaction will time out. The PCH will discard the cycle and set the DEV_ERR bit. The timeout minimum is 25 ms (800 RTC clocks). The Timeout counter inside the PCH will start after the first bit of data is transferred by the PCH and it is waiting for a response.

The 25 - ms Timeout counter will not count under the following conditions:

1. BYTE_DONE_STATUS bit (SMBus I/O Offset 00h, Bit 7) is set
2. The SECOND_TO_STS bit (TCO I/O Offset 06h, Bit 1) is not set (this indicates that the system has not locked up).

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.

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

Table 70. Enable for SMBALERT#

Event	INTREN (Host Control I/O Register, Offset 02h, Bit 0)	SMB_SMI_EN (Host Configuration Register, D31:F4:Offset 40h, Bit 1)	SMBALERT_DIS (Slave Command I/O Register, Offset 11h, Bit 2)	Result
SMBALERT# asserted low (always reported in Host Status Register, Bit 5)	X	X	X	Wake generated
	X	1	0	Slave SMI# generated (SMBUS_SMI_STS)
	1	0	0	Interrupt generated

Table 71. Enables for SMBus Slave Write and SMBus Host Events

Event	INTREN (Host Control I/O Register, Offset 02h, Bit 0)	SMB_SMI_EN (Host Configuration Register, D31:F4:Offset 40h, Bit 1)	Event
Slave Write to Wake/SMI# Command	X	X	Wake generated when asleep. Slave SMI# generated when awake (SMBUS_SMI_STS).
Slave Write to SMLINK_SLAVE_SMI Command	X	X	Slave SMI# generated when in the S0 state (SMBUS_SMI_STS)
Any combination of Host Status Register [4:1] asserted	0	X	None
	1	0	Interrupt generated
	1	1	Host SMI# generated

Table 72. Enables for the Host Notify Command

HOST_NOTIFY_INTREN (Slave Control I/O Register, Offset 11h, Bit 0)	SMB_SMI_EN (Host Configuration Register, D31:F4:Off40h, Bit 1)	HOST_NOTIFY_WKEN (Slave Control I/O Register, Offset 11h, Bit 1)	Result
0	X	0	None
X	X	1	Wake generated
1	0	X	Interrupt generated
1	1	X	Slave SMI# generated (SMBUS_SMI_STS)

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.

26.1.2 SMBus Slave Interface

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

Key features and capabilities include:

- Supports decode of three types of messages: Byte Write, Byte Read, and Host Notify.
- Receive Slave Address register: This is the address that the PCH decodes. A default value is provided so that the slave interface can be used without the processor having to program this register.
- Receive Slave Data register in the SMBus I/O space that includes the data written by the external micro controller.
- Registers that the external micro controller can read to get the state of the PCH.
  - Status bits to indicate that the SMBus slave logic caused an interrupt or SMI# Bit 0 of the Slave Status Register for the Host Notify command.
  - Bit 16 of the SMI Status Register for all others.

NOTE

The external micro controller should not attempt to access the PCH SMBus slave logic until either:

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

If a master leaves the clock and data bits of the SMBus interface at 1 for 50 µs or more in the middle of a cycle, the PCH slave logic's behavior is undefined. This is interpreted as an unexpected idle and should be avoided when performing management activities to the slave logic.

Format of Slave Write Cycle

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

The table below has the values associated with the registers.

Table 73. Slave Write Registers

Register	Function
0	Command Register. Refer to the table below for valid values written to this register.
1–3	Reserved
4	Data Message Byte 0
5	Data Message Byte 1
6–FFh	Reserved
Note: The external micro controller 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 74. Command Types

Command Type	Description
0	Reserved
1	**WAKE/SMI#.** This command wakes the system if it is not already awake. If system is already awake, an SMI# is generated.
2	**Unconditional Powerdown.** This command sets the PWRBTNOR_STS bit, and has the same effect as the Power button Override occurring.
3	**HARD RESET WITHOUT CYCLING:** This command causes a soft reset of the system (does not include cycling of the power supply). This is equivalent to a write to the CF9h register with Bits 2:1 set to 1, but Bit 3 set to 0.
4	**HARD RESET SYSTEM.** This command causes a hard reset of the system (including cycling of the power supply). This is equivalent to a write to the CF9h register with Bits 3:1 set to 1.
5	**Disable the TCO Messages.** This command will disable the PCH from sending Heartbeat and Event messages. Once this command has been executed, Heartbeat and Event message reporting can only be re - enabled by assertion and then de - assertion of the RSMRST# signal.
6	**WD RELOAD:** Reload watchdog timer.
7	Reserved
8	**SMLINK_SLV_SMI.** When the PCH detects this command type while in the S0 state, it sets the SMLINK_SLV_SMI_STS bit. This command should only be used if the system is in an S0 state. If the message is received during S4 and 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.

Format of Read Command

The external master performs Byte Read commands to the PCH SMBus Slave interface. The “Command” field (bits 18:11) indicate which register is being accessed. The Data field (bits 30:37) contain the value that should be read from that register.

Table 75. Slave Read Cycle Format

Bit	Description	Driven By	Comment
1	Start	External Micro controller	
2–8	Slave Address - 7 bits	External Micro controller	Must match value in Receive Slave Address register
9	Write	External Micro controller	Always 0
10	ACK	PCH	
11–18	Command code – 8 bits	External Micro controller	Indicates which register is being accessed. Refer to the Table below for a list of implemented registers.
19	ACK	PCH	
20	Repeated Start	External Micro controller	
21–27	Slave Address - 7 bits	External Micro controller	Must match value in Receive Slave Address register
28	Read	External Micro controller	Always 1
29	ACK	PCH	
30–37	Data Byte	PCH	Value depends on register being accessed. Refer to the Table below for a list of implemented registers.
38	NOT ACK	External Micro controller	
39	Stop	External Micro controller	

Table 76. Data Values for Slave Read Registers

Register	Bits	Description
0	7:0	Reserved
1	2:0	**System Power State** 000 = S0 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	Reserved
	2	Reserved
	3	1 = **SECOND_TO_STS** bit set. This bit will be set after the second Timeout (SECOND_TO_STS bit) of the Watchdog Timer occurs.

continued...

Register	Bits	Description
	6:4	Reserved. Will always be 0, but software should ignore.
	7	**SMBALERT# Status.** Reflects the value of the SMBALERT# pin (when the pin is configured to SMBALERT#). Valid only if SMBALERT_DISABLE = 0. Value always returns 1 if SMBALERT_DISABLE = 1.
5	0	**FWH bad bit.** This bit will be 1 to indicate that the FWH read returned FFh, which indicates that it is probably blank.
	1	**Battery Low Status.** 1 if the BATLOW# pin a low.
	2	**SYS_PWROK Failure Status:** This bit will be 1 if the SYSPWR_FLR bit in the GEN_PMCON_2 register is set.
	3	Reserved
	4	Reserved
	5	**POWER_OK_BAD:** Indicates the failure core power well ramp during boot/resume. This bit will be active if the SLP_S3# pin is de - asserted and PCH_PWROK pin is not asserted.
	6	**Thermal Trip:** This bit will shadow the state of processor Thermal Trip status bit (CTS). Events on signal will not create a event message
	7	Reserved: Default value is "X" *Note:* Software should not expect a consistent value when this bit is read through SMBUS/ SMLink
6	7:0	Contents of the Message 1 register.
7	7:0	Contents of the Message 2 register.
8	7:0	Contents of the WDSTATUS register.
9	7:0	Seconds of the RTC
A	7:0	Minutes of the RTC
B	7:0	Hours of the RTC
C	7:0	"Day of Week" of the RTC
D	7:0	"Day of Month" of the RTC
E	7:0	Month of the RTC
F	7:0	Year of the RTC
10h–FFh	7:0	Reserved

- **Behavioral Notes**

  According to SMBus protocol, Read and Write messages always begin with a Start bit—Address—Write bit sequence. When the PCH detects that the address matches the value in the Receive Slave Address register, it will assume that the protocol is always followed and ignore the Write bit (Bit 9) and signal an Acknowledge during bit 10. In other words, if a Start—Address—Read occurs (which is invalid for SMBus Read or Write protocol), and the address matches the PCH's Slave Address, the PCH will still grab the cycle.

  Also according to SMBus protocol, a Read cycle contains a Repeated Start—Address—Read sequence beginning at Bit 20. Once again, if the Address matches the PCH's Receive Slave Address, it will assume that the protocol is followed, ignore bit 28, and proceed with the Slave Read cycle.

Slave Read of RTC Time Bytes

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

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

The PCH SMBus slave interface only supports Byte Read operation. The external SMBus master will read the RTC time bytes one after another. It is the software's responsibility to check and manage the possible time rollover when subsequent time bytes are read.

For example, assuming the RTC time is 11 hours: 59 minutes: 59 seconds. When the external SMBus master reads the hour as 11, then proceeds to read the minute, it is possible that the rollover happens between the reads and the minute is read as 0. This results in 11 hours: 0 minute instead of the correct time of 12 hours: 0 minutes. Unless it is certain that rollover will not occur, software is required to detect the possible time rollover by reading multiple times such that the read time bytes can be adjusted accordingly if needed.

Format of Host Notify Command

The PCH tracks and responds to the standard Host Notify command as specified in the *System Management Bus (SMBus) Specification,* Version 2.0. The host address for this command is fixed to 0001000b. If the PCH already has data for a previously - received host notify command which has not been serviced yet by the host software (as indicated by the HOST_NOTIFY_STS bit), then it will NACK following the host address byte of the protocol. This allows the host to communicate non - acceptance to the master and retain the host notify address and data values for the previous cycle until host software completely services the interrupt.

NOTE

Host software must always clear the HOST_NOTIFY_STS bit after completing any necessary reads of the address and data registers.

The table below shows the Host Notify format:

Table 77. Host Notify Format

Bit	Description	Driven By	Comment
1	Start	External Master	
8:2	SMB Host Address – 7 bits	External Master	Always 0001_000
9	Write	External Master	Always 0
10	ACK (or NACK)	PCH	PCH NACKs if HOST_NOTIFY_STS is 1
17:11	Device Address – 7 bits	External Master	Indicates the address of the master; loaded into the Notify Device Address Register
18	Unused – Always 0	External Master	7 - bit - only address; this bit is inserted to complete the byte

continued...

Bit	Description	Driven By	Comment
19	ACK	PCH	
27:20	Data Byte Low – 8 bits	External Master	Loaded into the Notify Data Low Byte Register
28	ACK	PCH	
36:29	Data Byte High – 8 bits	External Master	Loaded into the Notify Data High Byte Register
37	ACK	PCH	
38	Stop	External Master	

Table 78. Data Values for Slave Read Registers

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 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 return 1 if SMBALERT_DISABLE = 1. (high = 1, low = 0).
5	0	**FWH bad bit:** This bit will be 1 to indicate that the FWH read returned FFh, which indicates that it is probably blank.
	1	**Battery Low Status:** 1 if the BATLOW# pin is a 0.
	2	**SYS_PWROK Failure Status:** This bit will be 1 if the SYSPWR_FLR bit in the GEN_PMCON_2 register is set.
	3	Reserved
	4	Reserved
	5	**POWER_OK_BAD:** Indicates the failure core power well ramp during boot/resume. This bit will be active if the PCH_PWROK pin is not asserted.

continued...

Register	Bits	Description
	6	**Thermal Trip:** This bit will shadow the state of processor Thermal Trip status bit (CTS). Events on signal will not create a event message.
	7	Reserved: Default value is "X" *Note:* Software should not expect a consistent value when this bit is read through SMBUS/ SMLink
6	7:0	Contents of the Message 1 register.
7	7:0	Contents of the Message 2 register.
8	7:0	Contents of the WDSTATUS register.
9	7:0	Seconds of the RTC
A	7:0	Minutes of the RTC
B	7:0	Hours of the RTC
C	7:0	"Day of Week" of the RTC
D	7:0	"Day of Month" of the RTC
E	7:0	Month of the RTC
F	7:0	Year of the RTC
10h–FFh	7:0	Reserved

Table 79. Enables for SMBus Slave Write and SMBus Host Events

Event	INTREN (Host Control I/O Register, Offset 02h, Bit 0)	SMB_SMI_EN (Host Configuration Register, D31:F3:Offset 40h, Bit 1)	Event
Slave Write to Wake/SMI# Command	X	X	Wake generated when asleep. Slave SMI# generated when awake (SMBUS_SMI_STS)
Slave Write to SMLINK_SLAVE_SMI Command	X	X	Slave SMI# generated when in the S0 state (SMBUS_SMI_STS)
Any combination of Host Status Register [4:1] asserted	0	X	None
	1	0	Interrupt generated
	1	1	Host SMI# generated

26.2 Signal Description

Signal Name	Type	Description
GPP_C0 / **SMBCLK**	I/OD	**SMBus Clock.** External Pull-up resistor is required.
GPP_C1 / **SMBDATA**	I/OD	**SMBus Data.** External Pull-up resistor is required.
GPP_C2 / **SMBALERT#**	I/OD	**SMBus Alert:** This signal is used to wake the system or generate SMI#. External Pull-up resistor is required.

26.3 Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value	Notes
SMBALERT#	Pull-down	20 kohm ± 30%	The internal pull-down resistor is disable after RSMRST# de-asserted.

26.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
SMBDATA	Primary	Undriven	Undriven	Undriven	Undriven
SMBCLK	Primary	Undriven	Undriven	Undriven	Undriven
SMBALERT#	Primary	Undriven	Undriven	Undriven	OFF
Note: 1. Reset reference for primary well pins is RSMRST#.					

26.5 SMBus Power Gating

SMBus shares the Power Gating Domain with Primary-to-Sideband Bridge (P2SB). A single FET controls the single Power Gating Domain; but SMBus and P2SB each has its own dedicated Power Gating Control Block. The FET is only turned off when all these interfaces are ready to PG entry or already in the PG state.

27.0 Serial Peripheral Interface (SPI)

The PCH provides two Serial Peripheral Interfaces (SPI). The SPI0 interface consists of three Chip Select signals. SPI0 interface can allow two flash memory devices (SPI0_CS0# and SPI0_CS1#) and one TPM device (SPI0_CS2#) to be connected to the PCH. The SPI0 interface support either 1.8 V or 3.3 V. The voltage is selected via a Hardware strap on SPIVCCIOSEL signal. Refer to VCCSPI Voltage (3.3 V or 1.8 V) Selection on page 179.

Table 80. Acronyms

Acronyms	Description
CLK	Clock
CS	Chip Select
FCBA	Flash Component Base Address
FIBA	Flash Initialization Base Address
FLA	Flash Linear Address
FMBA	Flash Master Base Address
FPSBA	Flash PCH Strap Base Address
FRBA	Flash Region Base Address
MDTBA	MIP Descriptor Table Base Address
MISO	Mater In Slave Out
MOSI	Mater Out Slave In
TPM	Trusted Platform Module

27.1 Functional Description

This section provides information on the following topics:

- SPI0 for Flash
- SPI0 support for TPM

27.1.1 SPI0 for Flash

The Serial Peripheral Interface (SPI0) supports two SPI flash devices via two chip select (SPI0_CS0# and SPI0_CS1#). The maximum size of flash supported is determined by the SFDP-discovered addressing capability of each device. Each component can be up to 16 MB (32 MB total addressable) using 3-byte addressing. Each component can be up to 64 MB (128 MB total addressable) using 4-byte addressing. Another chip select (SPI0_CS2#) is also available and only used for TPM on SPI support. PCH drives the SPI0 interface clock at either 14 MHz, 25 MHz, 33 MHz, or 50 MHz and will function with SPI flash/TPM devices that support at least one of these frequencies. The SPI interface supports either 3.3 V or 1.8 V.

A SPI0 flash device supporting SFDP (Serial Flash Discovery Parameter) is required for all PCH design. A SPI0 flash device on SPI0_CS0# with a valid descriptor MUST be attached directly to the PCH.

The PCH supports fast read which consist of:

1. Dual Output Fast Read (Single Input Dual Output)
2. Dual I/O Fast Read (Dual Input Dual Output)
3. Quad Output Fast Read (Single Input Quad Output)
4. Quad I/O Fast Read (Quad Input Quad Output)

The PCH SPI0 has a third chip select SPI0_CS2# for TPM support over SPI. The TPM on SPI0 will use SPI0_CLK, SPI0_MISO, SPI0_MOSI and SPI0_CS2# SPI signals.

SPI0 Supported Features

- **Descriptor Mode**

  Descriptor Mode is required. Non-Descriptor Mode is not supported.

- **SPI0 Flash Regions**

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

Table 81. SPI0 Flash Regions

Region	Content
0	Flash Descriptor
1	BIOS
2	Intel® Converged Security and Management Engine
3	Gigabit Ethernet
4	Platform Data
5	EC

Only four masters can access the regions: Host processor running BIOS code, Integrated Gigabit Ethernet and Host processor running Gigabit Ethernet Software, Intel® Converged Security and Management Engine, and the EC.

The Flash Descriptor and Intel® CSME region are the only required regions. The Flash Descriptor has to be in region 0 and region 0 must be located in the first sector of Device 0 (Offset 0). All other regions can be organized in any order.

Regions can extend across multiple components, but must be contiguous.

Flash Region Sizes

SPI0 flash space requirements differ by platform and configuration. The Flash Descriptor requires one 4 KB or larger block. GbE requires two 4 KB or larger blocks. The amount of flash space consumed is dependent on the erase granularity of the flash part and the platform requirements for the Intel® CSME and BIOS regions. The Intel® CSME region contains firmware to support Intel Active Management Technology and other Intel® CSME capabilities.

Table 82. Region Size Versus Erase Granularity of Flash Components

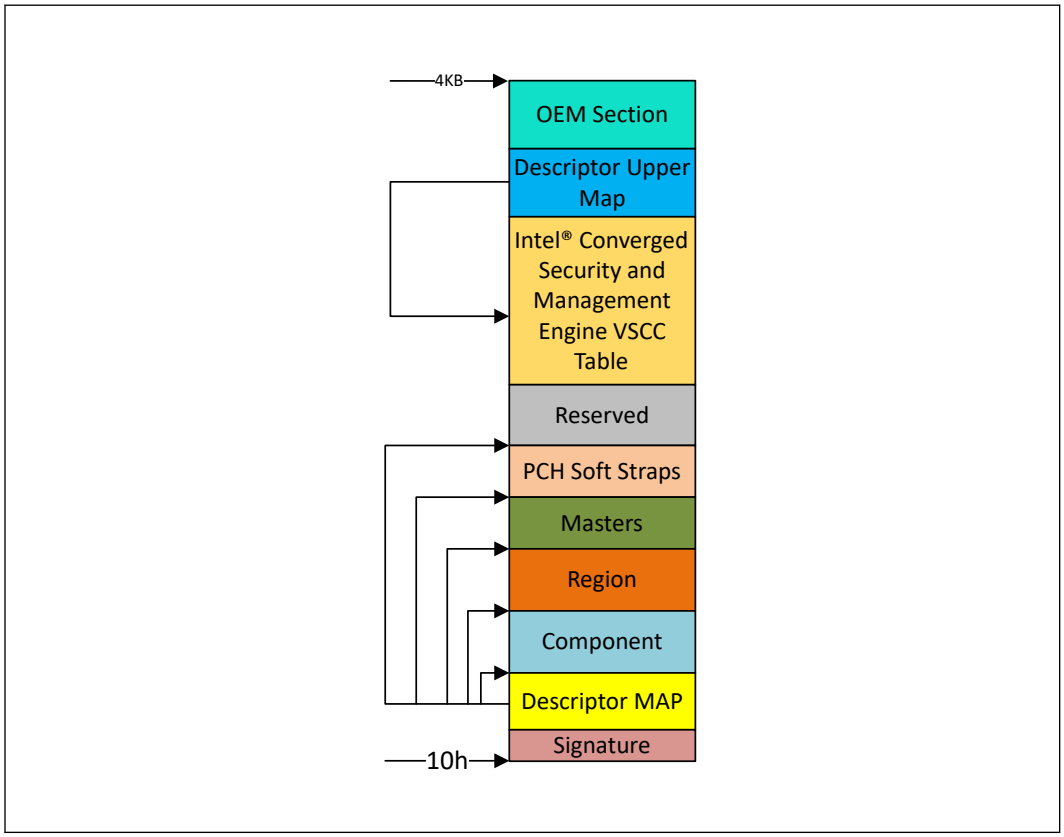
Region	Size with 4 KB Blocks	Size with 8 KB Blocks	Size with 64 KB Blocks
Descriptor	4 KB	8 KB	64 KB
GbE	8 KB	16 KB	128 KB
BIOS	Varies by Platform	Varies by Platform	Varies by Platform
Intel® CSME	Varies by Platform	Varies by Platform	Varies by Platform
EC	Varies by Platform	Varies by Platform	Varies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

Flash Descriptor

The bottom sector of the flash component 0 contains the Flash Descriptor. The maximum size of the Flash Descriptor is 4 KB. If the block/sector size of the SPI0 flash device is greater than 4 KB, the flash descriptor will only use the first 4 KB of the first block. The flash descriptor requires its own block at the bottom of memory (00h). The information stored in the Flash Descriptor can only be written during the manufacturing process as its read/write permissions must be set to read only when the computer leaves the manufacturing floor.

The Flash Descriptor is made up of eleven sections as shown in the figure below:

Figure 17. Flash Descriptor Regions



- The Flash signature selects Descriptor Mode as well as verifies if the flash is programmed and functioning. The data at the bottom of the flash (offset 10h) must be 0FF0A55Ah in order to be in Descriptor mode.
- The Descriptor map has pointers to the other five descriptor sections as well as the size of each.
- The component section has information about the SPI0 flash in the system including: the number of components, density of each, invalid instructions (such as chip erase), and frequencies for read, fast read and write/erase instructions.
- The Region section points to the three other regions as well as the size of each region.
- The master region contains the security settings for the flash, granting read/write permissions for each region and identifying each master by a requester ID.
- The processor and PCH Soft Strap sections contain processor and PCH configurable parameters.
- The Reserved region between the top of the processor strap section and the bottom of the OEM Section is reserved for future chipset usages.
- The Descriptor Upper MAP determines the length and base address of the Intel® Converged Security and Management Engine VSCC Table.
- The Intel® Converged Security and Management Engine VSCC Table holds the JEDEC ID and the VSCC information of the entire SPI0 Flash supported by the NVM image.
- OEM Section is 256 bytes reserved at the top of the Flash Descriptor for use by OEM.
- **Descriptor Master Region**

  The master region defines read and write access setting for each region of the SPI0 device. The master region recognizes four masters: BIOS, Gigabit Ethernet, Intel® Converged Security and Management Engine, and EC. Each master is only allowed to do direct reads of its primary regions.

Table 83. Region Access Control Table

Master Read/Write Access				
Region	**Processor and BIOS**	**Intel® CSME**	**GbE Controller**	**EC**
BIOS	Read/Write	N/A	Read/Write	Note
Intel®Converged Security Management Engine (CSME)	N/A	Read/Write	Read	N/A
Gigabit Ethernet	N/A	N/A	Read/Write	N/A
EC	Read	N/A	N/A	Read/Write
Note: Optional BIOS access to the EC region.				

- **Flash Descriptor CPU Complex Soft Strap Section**

Region Name	Starting Address
Signature	10h
Component FCBA	30h
Regions FRBA	40h

continued...

Region Name	Starting Address
Masters FMBA	80h
PCH Straps FPSBA	100h
MDTBA	C00h
PMC Straps	C14h
CPU Straps	C2Ch
Intel® CSME Straps	C3Ch
Register Init FIBA	340h

Flash Access

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

- **Direct Access**
  - Masters are allowed to do direct read only of their primary region
    - Gigabit Ethernet region can only be directly accessed by the Gigabit Ethernet controller. Gigabit Ethernet software must use Program Registers to access the Gigabit Ethernet region.
  - Master's Host or Intel® Converged Security and Management Engine virtual read address is converted into the SPI0 Flash Linear Address (FLA) using the Flash Descriptor Region Base/Limit registers

    Direct Access Security
  - Requester ID of the device must match that of the primary Requester ID in the Master Section
  - Calculated Flash Linear Address must fall between primary region base/limit
  - Direct Write not allowed
  - Direct Read Cache contents are reset to 0's on a read from a different master
- **Program Register Access**
  - Program Register Accesses are not allowed to cross a 4 KB boundary and can not issue a command that might extend across two components
  - Software programs the FLA corresponding to the region desired
    - Software must read the devices Primary Region Base/Limit address to create a FLA.

  Register Access Security
  - Only primary region masters can access the registers

27.1.2 SPI0 Support for TPM

The PCH's SPI0 flash controller supports a discrete TPM on the platform via its dedicated SPI0_CS2# signal. The platform must have no more than 1 TPM.

SPI0 controller supports accesses to SPI0 TPM at approximately 17 MHz, 33 MHz and 48 MHz depending on the PCH soft strap. 20 MHz is the reset default, a valid PCH soft strap setting overrides the requirement for the 20 MHz. SPI0 TPM device must support a clock of 20 MHz, and thus should handle 15-20 MHz. It may but is not required to support a frequency greater than 20 MHz.

TPM requires the support for the interrupt routing. However, the TPM's interrupt pin is routed to the PCH's interrupt configurable GPIO pin. Thus, TPM interrupt is completely independent from the SPI0 controller.

27.2 Signal Description

Signal Name	Type	Description
SPI0_CLK	O	**SPI0 Clock**: SPI clock signal for the common flash/TPM interface. Supports 20 MHz, 33 MHz and 50 MHz.
SPI0_CS0#	O	**SPI0 Chip Select 0**: Used to select the primary SPI0 Flash device. *Note:* This signal cannot be used for any other type of device than SPI Flash.
SPI0_CS1#	O	**SPI0 Chip Select 1**: Used to select an optional secondary SPI0 Flash device. *Note:* This signal cannot be used for any other type of device than SPI Flash.
SPI0_CS2#	O	**SPI0 Chip Select 2**: Used to select the TPM device if it is connected to the SPI0 interface. It cannot be used for any other type of device.
SPI0_MOSI	I/O	**SPI0 Master OUT Slave IN**: Defaults as a data output pin for PCH in Dual Output Fast Read mode. Can be configured with a Soft Strap as a bidirectional signal (SPI0_IO0) to support the Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes.
SPI0_MISO	I/O	**SPI0 Master IN Slave OUT**: Defaults as a data input pin for PCH in Dual Output Fast Read mode. Can be configured with a Soft Strap as a bidirectional signal (SPI0_IO1) to support the Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes.
SPI0_IO2	I/O	**SPI0 Data I/O**: A bidirectional signal used to support Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes. This signal is not used in Dual Output Fast Read mode.
SPI0_IO3	I/O	**SPI0 Data I/O**: A bidirectional signal used to support Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes. This signal is not used in Dual Output Fast Read mode.

27.3 Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value	Notes
SPI0_CLK	Pull-down	20 kohm ± 30%	
SPI0_MOSI	Pull-up	20 kohm ± 30%	Note
SPI0_MISO	Pull-up	20 kohm ± 30%	Note
SPI0_CS[2:0]#	Pull-down	20 kohm ± 30%	
SPI0_IO[2:3]	Pull-up	20 kohm ± 30%	Note

NOTE

The internal pull-up is disabled when RSMRST# is asserted (during reset) and only enabled after RSMRST# de-assertion.

27.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
SPI0_CLK	Primary	Internal Pull-down	Driven Low	Driven Low	OFF
SPI0_MOSI	Primary	Hi-Z (Refer to Note 2)	Internal PU, then Driven Low	Driven Low	OFF
SPI0_MISO	Primary	Hi-Z	Internal Pull-up	Internal Pull-up	OFF
SPI0_CS0#	Primary	Internal Pull-down	Driven High	Driven High	OFF
SPI0_CS1#	Primary	Internal Pull-down	Driven High	Driven High	OFF
SPI0_CS2#	Primary	Internal Pull-down	Driven High	Driven High	OFF
SPI0_IO[3:2]	Primary	Hi-Z (Refer to Note 2)	Internal Pull-up	Internal Pull-up	OFF

Notes: 1. During reset refers to when RSMRST# is asserted.
2. SPI0_MOSI, SPI0_IO[3:2] also function as strap pins. The actual pin state during Reset is dependent on the platform Pull-up/Pull-down resistor.

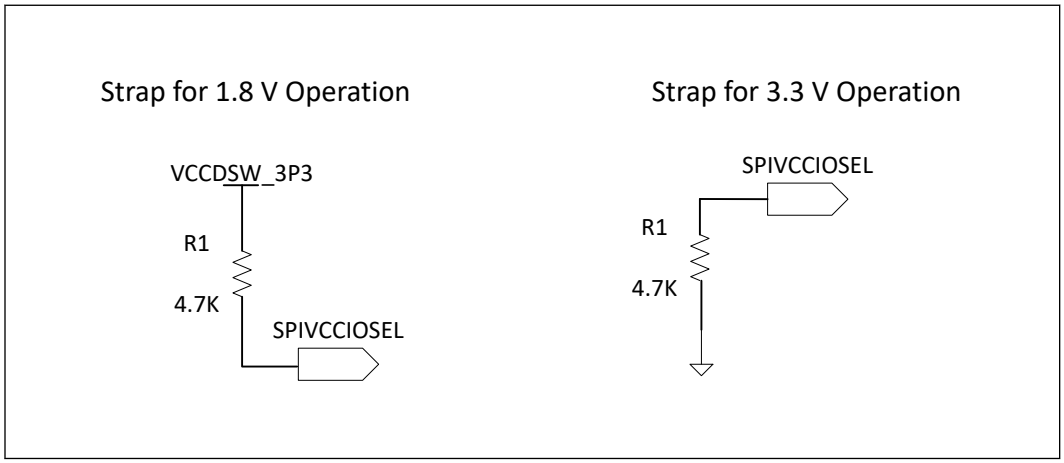
27.5 VCCSPI Voltage (3.3 V or 1.8 V) Selection

The VCCSPI voltage (3.3 V or 1.8 V) is selected via a strap on SPIVCCIOSEL.

0 = SPI voltage is 3.3 V (4.7 kohm pull-down to GND)

1 = SPI voltage is 1.8 V (4.7 kohm pull-up to VCCDSW_3P3)

Figure 18. VCCSPI Voltage (3.3 V or 1.8 V) Selection



28.0 Touch Host Controller (THC)

Touch Host Controller provides the standard SPI interface for SoCs to connect to external touch ICs. In the first generation THC, only SPI IOs are supported.

THC also supports the GPIO based SPI interrupt from touch IC, and supports hardware autonomous power management scheme within the SoC.

Table 84. Acronyms

Acronyms	Description
CLK	Clock
CS	Chip Select
MISO	Master In Slave Out
MOSI	Master Out Slave In
TPM	Trusted Platform Module

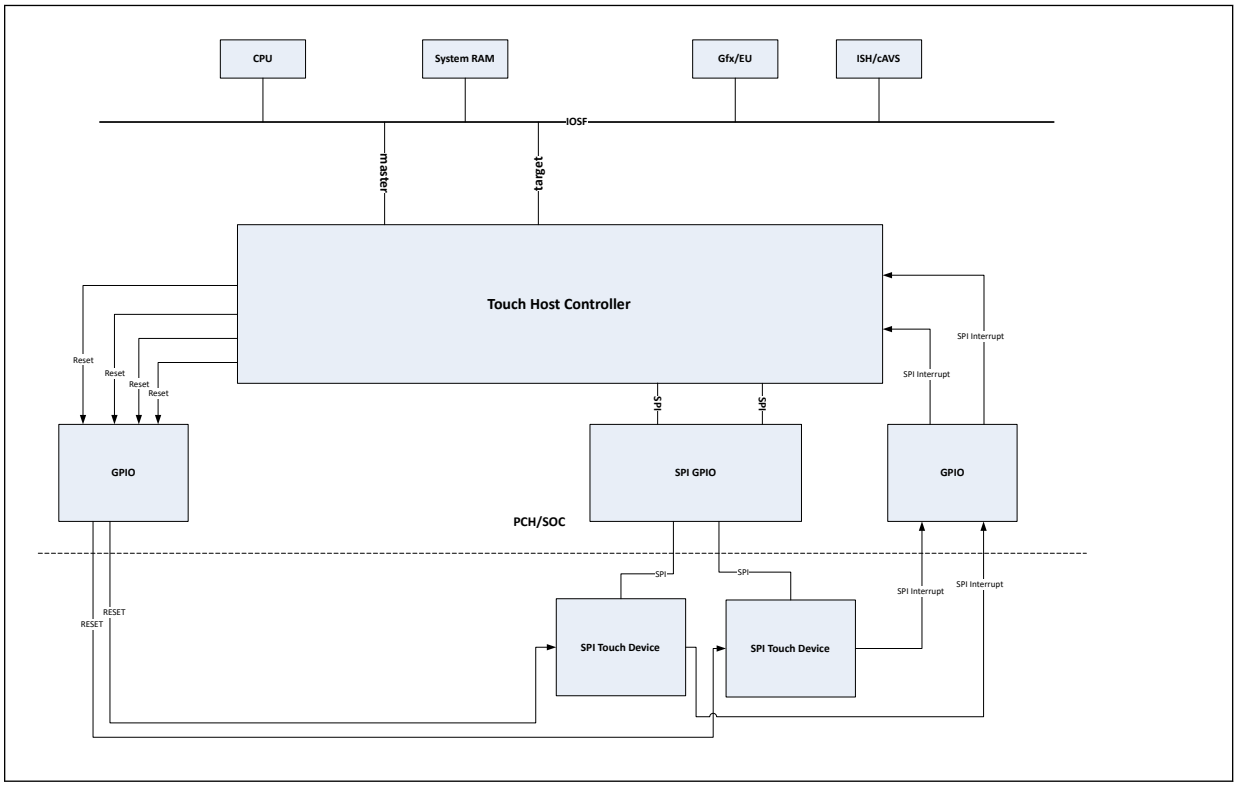
28.1 Functional Description

The Touch Host Controller (THC) supports a host controller interface to the touch IC for high bandwidth touch data transfer from SPI based touch ICs. THC provides high bandwidth DMA services to the touch driver and transfer the touch raw data or HID reports to internal touch accelerator (example, graphics EUs or host CPU), or host driver respectively.

The THC controller bridges the SoC bus and SPI ports, below are the details.

- THC Controller
  - Touch Host controller bridges the SoC bus and SPI
  - The THC Controller has the following interfaces
    - IOSF Primary Interface for DMA operation and register access
      - Minimum 100MHz 64 bit
    - SPI IO interface
- SPI IO
  - 1.8V SPI IOs
  - Provides SPI interface to the THC core

Figure 19. THC Block Diagram



28.2 Signal Description

Signal Name	Type	Description
GPP_E11 / **THC0_SPI1_CLK** / GSPI0_CLK	O	**THC0_SPI1 Clock**: THC SPI1 clock output from PCH. Supports 20 MHz, 33 MHz and 50 MHz.
GPP_F11 / **THC1_SPI2_CLK** / GSPI1_CLK	O	**THC1_SPI2 Clock**: THC SPI2 clock output from PCH.
GPP_E10 / **THC0_SPI1_CS#** / GSPI0_CS0#	O	**THC0_SPI1 Chip Select**: Used to select the touch devices if it is connected to THC0_SPI1 interface.
GPP_F16 / GSXCLK / **THC1_SPI2_CS#** / GSP1_CS0#	O	**THC1_SPI2 Chip Select**: Used to select the touch devices if it is connected to THC1_SPI2 interface.
GPP_E13 / **THC0_SPI1_IO0** / GSPI0_MOSI	I/O	**THC0_SPI1_IO0**: A bidirectional signal used to support single, dual and quad mode data transfer.
GPP_E12 / **THC0_SPI1_IO1** / GSPI0_MISO	I/O	**THC0_SPI1_IO1**: A bidirectional signal used to support single, dual and quad mode data transfer.
GPP_E1 / **THC0_SPI1_IO2**	I/O	**THC0_SPI1_IO2**: A bidirectional signal used to support single, dual and quad mode data transfer.

continued...

Signal Name	Type	Description
GPP_E2 / **THC0_SPI1_IO3**	I/O	**THC0_SPI1_IO3**: A bidirectional signal used to support single, dual and quad mode data transfer.
GPP_F12 / GSXDOUT / **THC1_SPI2_IO0** / GSPI1_MOSI	I/O	**THC1_SPI2_IO0**: A bidirectional signal used to support single, dual and quad mode data transfer.
GPP_F13 / GSXSLOAD / **THC1_SPI2_IO1** / GSPI1_MISO	I/O	**THC1_SPI2_IO1**: A bidirectional signal used to support single, dual and quad mode data transfer.
GPP_F14 / GSXDIN / **THC1_SPI2_IO2**	I/O	**THC1_SPI2_IO2**: A bidirectional signal used to support single, dual and quad mode data transfer.
GPP_F15 / GSXSRESET# / **THC1_SPI2_IO3**	I/O	**THC1_SPI2_IO3**: A bidirectional signal used to support single, dual and quad mode data transfer.
GPP_E6 / **THC0_SPI1_RST#**	O	**THC0_SPI1 Reset:** THC0_SPI1 Reset signal from Touch host controller.
GPP_F17 / **THC1_SPI2_RST#**	O	**THC1 SPI2 Reset:** THC1_SPI2 Reset signal from Touch host controller.
GPP_E17 / **THC0_SPI1_INT#**	I	**THC0 SPI1 interrupt:** THC0_SPI1 Interrupt signal.
GGPP_F18 / **THC1_SPI2_INT#**	I	**THC1 SPI2 interrupt:** THC1_SPI2 Interrupt signal.

28.3 Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value	Notes
THC0_SPI1_IO[0:3]	Pull-up	20 kohm ± 30%	
THC1_SPI2_IO[0:3]	Pull-up	20 kohm ± 30%	

NOTE

The internal pull-up is disabled when RSMRST# is asserted (during reset).

28.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
THC0_SPI1_CLK	Primary	Undriven	Undriven	Undriven	OFF
THC1_SPI2_CLK	Primary	Undriven	Undriven	Undriven	OFF
THC0_SPI1_CS#	Primary	Undriven	Undriven	Undriven	OFF
THC1_SPI2_CS#	Primary	Undriven	Undriven	Undriven	OFF
THC0_SPI1_IO[0:3]	Primary	Undriven	Undriven	Undriven	OFF
THC1_SPI2_IO[0:3]	Primary	Undriven	Undriven	Undriven	OFF
THC0_SPI1_RST#	Primary	Undriven	Undriven	Undriven	OFF
THC1_SPI2_RST#	Primary	Undriven	Undriven	Undriven	OFF

continued...

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
THC0_SPI1_INT#	Primary	Undriven	Undriven	Undriven	OFF
THC1_SPI2_INT#	Primary	Undriven	Undriven	Undriven	OFF
Note: 1. During reset refers to when RSMRST# is asserted.					

29.0 Intel® Serial IO Generic SPI (GSPI) Controllers

The PCH implements three generic SPI interfaces to support devices that uses serial protocol for transferring data.

Each interface consists of a clock (CLK), two chip selects (CS) and two data lines (MOSI and MISO).

The GSPI interfaces support the following features:

- Support bit rates up to 20 Mbits/s
- Support data size from 4 to 32 bits in length and FIFO depths of 64 entries
- Support DMA with 128-byte FIFO per channel (up to 64-byte burst)
- Full duplex synchronous serial interface
- Support the Motorola's* SPI protocol
- Operate in master mode only

NOTE

Slave mode is not supported.

Table 85. Acronyms

Acronyms	Description
GSPI	Generic Serial Peripheral Interface
LTR	Latency Tolerance Reporting

29.1 Functional Description

This section provides information on the following topics:

- Controller Overview
- DMA Controller
- Reset
- Power Management
- Interrupts
- Error Handling

29.1.1 Controller Overview

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

The processor or DMA accesses data through the GSPI port's transmit and receive FIFOs.

A processor access takes the form of programmed I/O, transferring one FIFO entry per access. Processor accesses must always be 32 bits wide. Processor writes to the FIFOs are 32 bits wide, but the PCH will ignore all bits beyond the programmed FIFO data size. Processor reads to the FIFOs are also 32 bits wide, but the receive data written into the Receive FIFO is stored with '0' in the most significant bits (MSB) down to the programmed data size.

The FIFOs can also be accessed by DMA, which must be in multiples of 1, 2, or 4 bytes, depending upon the EDSS value, and must also transfer one FIFO entry per access.

For writes, the Enhanced SPI takes the data from the transmit FIFO, serializes it, and sends it over the serial wire to the external peripheral. Receive data from the external peripheral on the serial wire is converted to parallel words and stored in the receive FIFO.

A programmable FIFO trigger threshold, when exceeded, generates an interrupt or DMA service request that, if enabled, signals the processor or DMA respectively to empty the Receive FIFO or to refill the Transmit FIFO.

The GSPI controller, as a master, provides the clock signal and controls the chip select line. Commands codes as well as data values are serially transferred on the data signals. The PCH asserts a chip select line to select the corresponding peripheral device with which it wants to communicate. The clock line is brought to the device whether it is selected or not. The clock serves as synchronization of the data communication.

29.1.2 DMA Controller

The GSPI controllers have an integrated DMA controller.

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.

Channel Control

- The source transfer width and destination transfer width are programmable. The width can be programmed to 1, 2, or 4 bytes.

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

29.1.3 Reset

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

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

29.1.4 Power Management

Device Power Down Support

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

Latency Tolerance Reporting (LTR)

Latency Tolerance Reporting is used to allow the system to optimize internal power states based on dynamic data, comprehending the current platform activity and service latency requirements. However, the GSPI bus architecture does not provide the architectural means to define dynamic latency tolerance messaging. Therefore, the interface supports this by reporting its service latency requirements to the platform power management controller via LTR registers.

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

1. Platform/HW Default Control. This scheme is used for usage models in which the controller’s state correctly informs the platform of the current latency requirements. In this scheme, the latency requirement is a function of the controller state. The latency for transmitting data to/from its connected device at

a given rate while the controller is active is representative of the active latency requirements. On the other hand if the device is not transmitting or receiving data and idle, there is no expectation for end to end latency.

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

29.1.5 Interrupts

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

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

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

29.1.6 Error Handling

Errors that might occur on the external GSPI signals are comprehended by the host controller and reported to the interface host controller driver through the MMIO registers.

29.2 Signal Description

Signal Name	Type	Description
GPP_E10 / THC0_SPI1_CS# / **GSPI0_CS0#**	O	**Generic SPI 0 Chip Select**
GPP_E11 / THC0_SPI1_CLK / **GSPI0_CLK**	O	**Generic SPI 0 Clock**
GPP_E12 / THC0_SPI1_IO1 / I2C0A_SDA / **GSPI0_MISO**	I	**Generic SPI 0 MISO**
GPP_E13 / THC0_SPI1_IO0 / I2C0A_SCL / **GSPI0_MOSI**	O	**Generic SPI 0 MOSI**
GPP_F16 / GSXCLK / THC1_SPI2_CS# / **GSPI1_CS0#**	O	**Generic SPI 1 Chip Select 0**
GSPI1_CS1#/GPP_B23 / SML1ALERT# / PCHHOT#	O	**Generic SPI 1 Chip Select 1**
GPP_F11 / THC1_SPI2_CLK / **GSPI1_CLK**	O	**Generic SPI 1 Clock**
GPP_F13 / GSXSLOAD / THC1_SPI2_IO1 / **GSPI1_MISO**	I	**Generic SPI 1 MISO**
GPP_F12 / GSXDOUT / THC1_SPI2_IO0 / **GSPI1_MOSI**	O	**Generic SPI 1 MOSI**

continued...

Signal Name	Type	Description
GPP_D9 / ISH_SPI_CS# / DDP3_CTRLCLK / TBT_LSX2_TXD / BSSB_LS2_RX / **GSPI2_CS0#**	O	**Generic SPI 2 Chip Select 0**
GSPI2_CS1# / ISH_UART0_RTS#/GPP_D15 / IMGCLKOUT5	O	**Generic SPI 2 Chip Select 1**
GPP_D10 / ISH_SPI_CLK / DDP3_CTRLDATA / TBT_LSX2_RXD / BSSB_LS2_TX / **GSPI2_CLK**	O	**Generic SPI 2 Clock**
GPP_D11 / ISH_SPI_MISO / DDP4_CTRLCLK / TBT_LSX3_TXD / BSSB_LS3_RX / **GSPI2_MISO**	I	**Generic SPI 2 MISO**
GPP_D12 / ISH_SPI_MOSI / DDP4_CTRLDATA / TBT_LSX3_RXD / BSSB_LS3_TX / **GSPI2_MOSI**	O	**Generic SPI 2 MOSI**

29.3 Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value	Notes
GSPI0_MOSI	Pull-Down	20 kohm± 30%	The integrated pull-down is disabled after PCH_PWROK assertion
GSPI1_MOSI	Pull-Down	20 kohm ± 30%	The integrated pull-down is disabled after PCH_PWROK assertion
GSPI2_MOSI	Pull-Down	20 kohm ± 30%	The integrated pull-down is disabled after PCH_PWROK assertion
GSPI0_MISO	Pull-Down	20 kohm ± 30%	
GSPI1_MISO	Pull-Down	20 kohm ± 30%	
GSPI2_MISO	Pull-Down	20 kohm ± 30%	

29.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
GSPI0_CS0#, GSPI1_CS0#	Primary	Undriven	Undriven	Undriven	OFF
GSPI2_CLK, GSPI1_CLK, GSPI0_CLK	Primary	Undriven	Undriven	Undriven	OFF
GSPI2_MISO, GSPI1_MISO, GSPI0_MISO	Primary	Undriven	Undriven	Undriven	OFF
GSPI2_MOSI, GSPI1_MOSI, GSPI0_MOSI	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
Note: 1. Reset reference for primary well pins is RSMRST#.					

30.0 Testability

JTAG:

This section contains information regarding the testability signals that provides access to JTAG, run control, system control, and observation resources. JTAG (TAP) ports are compatible with the IEEE Standard Test Access Port and Boundary Scan Architecture 1149.1 and 1149.6 Specification. JTAG Pin definitions are from IEEE Standard Test Access Port and Boundary Scan. Architecture (IEEE Std. 1149.1-2001).

Intel® Trace Hub:

Intel® Trace Hub is a debug architecture that unifies hardware and software system visibility. Intel® Trace Hub is not merely intended for hardware debug or software debug, but full system debug. This includes debugging hardware and software as they interact and produce complex system behavior. Intel® Trace Hub defines new features and also leverages some existing debug technologies to provide a complete framework for hardware and software co-debug, software development and tuning, as well as overall system performance optimization.

There are multiple destinations to receive the trace data from Intel® Trace Hub:

- Direct Connect Interface (DCI)
  - — OOB Hosting DCI
  - — USB 3.2 hosting DCI.DBC
- System Memory

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

- BIOS
- Intel® CSME
- AET (Architecture Event Trace)
- Power Management Event Trace
- Windows* ETW (for driver or application)

Table 86. Acronyms

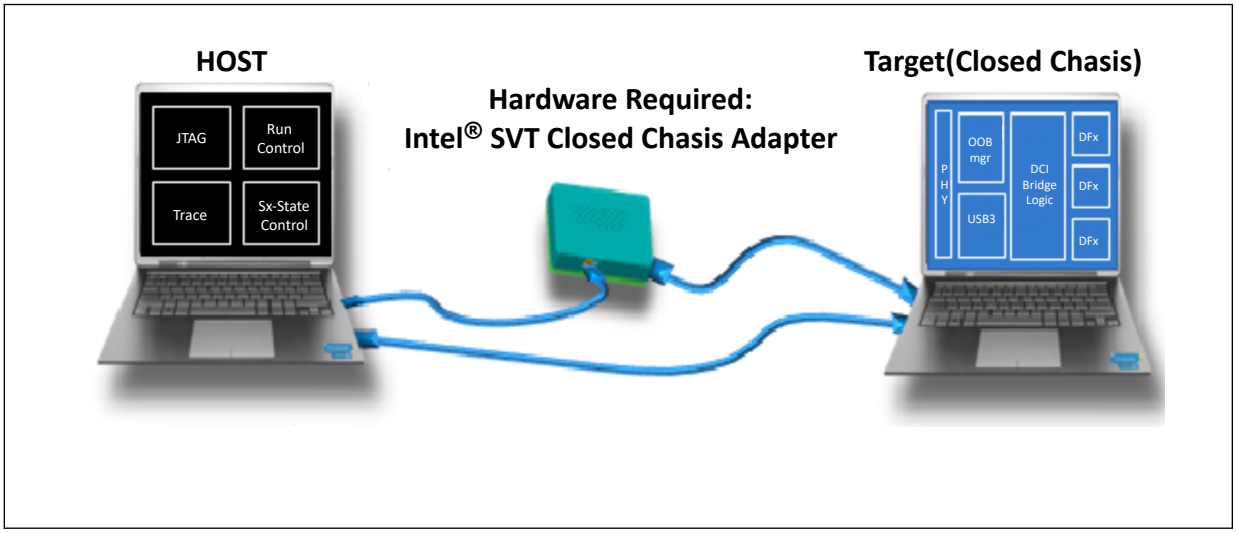
Acronyms	Description
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input/Output
I/OD	Input/Output Open Drain
JTAG	Joint Test Action Group
DCI	Direct Connect Interface
DbC	Debug Class Devices

Table 87. References

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

30.1 Intel® Trace Hub (Intel® TH)

Figure 20. Platform Setup with Intel® Trace Hub



30.2 Intel® Direct Connect Interface (DCI)

Direct Connect Interface (DCI) is a new debug transport technology to enable closed chassis debug through any of USB 3.2 ports out from Intel silicon. Some bridging logic is embedded in the silicon to "bridge" the gap between standard I/O ports and the debug interfaces including JTAG, probe mode, hooks, trace infrastructure, and etc. To control the operation of this embedded logic, a DCI packet based protocol is invented which controls and data can be sent or received. This protocol can operate over a few different physical transport paths to the target which known as "hosting interfaces".

NOTE

DCI and USB 3.2 based debugger (kernel level debugger) are mutually exclusive.

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

- OOB Hosting DCI
- USB 3.2 Hosting DCI.DBC

Supported capabilities in DCI are:

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

30.2.1 Out Of Band (OOB) Hosting DCI

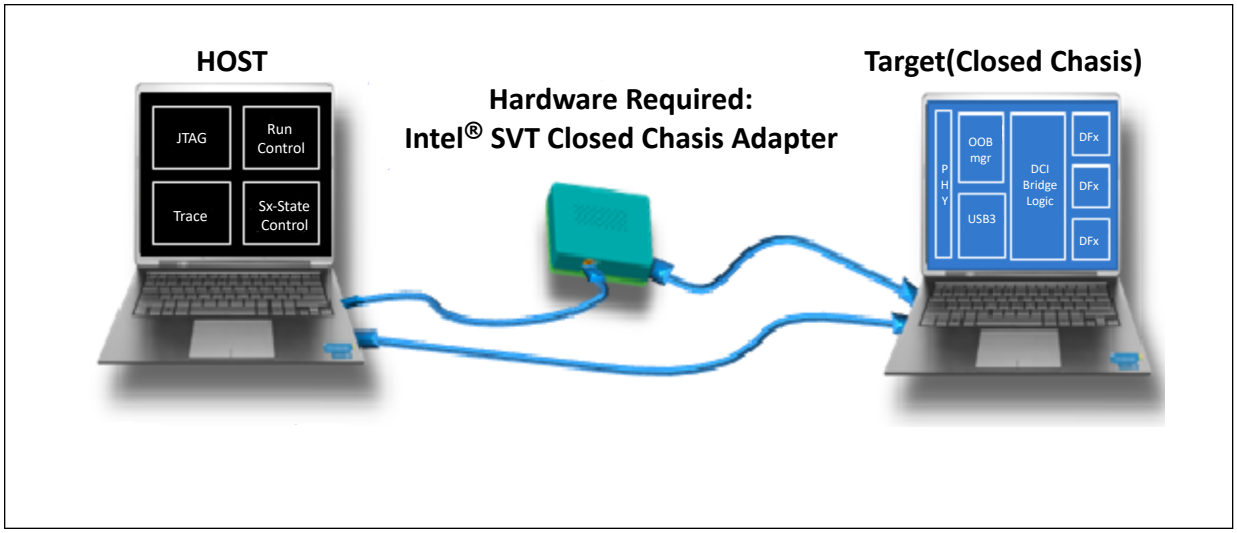
OOB was developed to provide an alternate path to convey controls and data to or from the EXI/DCI by connecting physically to the target through a USB 3.2 Gen 2x1 port. OOB provides an alternate side band path around the USB 3.2 controller, so that the embedded logic can be accessed, even when the USB 3.2 controller is not alive (such as in low power states) or is malfunctioning. This path does not rely on USB 3.2 Gen 2x1 protocol, link layer, or physical layer, because the xHCI functions are generally not available in such conditions. Instead, this path relies on a special adapter that was developed by Intel called the Intel® SVT Closed Chassis Adapter (CCA). It is a simple data transformation device. This adapter generates a OOB signaling protocol operating at up to 400 MHz and serializes data flowing through it. This adapter works together with debug host software and the embedded logic, contain a back-pressure scheme that makes both sides tolerant of overflow and starvation conditions, which is equivalent of USB 3.2 link layer. This path also uses native DCI packet protocol instead of USB 3.2 Gen 2x1 protocol. DCI.OOB - slower speed, CCA box needed. But survives S0ix and Sx states. Provides early boot access. Cannot tolerate re-driver circuits in its path.

30.2.2 USB 3.2 Hosting DCI.DBC

It relies on Debug Class Devices (DbC) which is comprised of a set of logic that is bolted to the side of the xHCI host controller and enable the target to act the role of a USB device for debug purpose. This path uses the USB packet protocol layer, USB layer flow control and USB physical layer at 5 GHz (for USB 3.2) and 480 MHz (for USB 2.0). DCI.DBC - Fast speed. USB 3.2 only works in S0. USB 2.0 survives S0ix and Sx states and provides early boot access.

30.2.3 Platform Setup

Figure 21. Platform Setup with DCI Connection



30.3 JTAG

This section provides information about Signal description and I/O Signal Planes and States.

30.3.1 Signal Description

Table 88. Testability Signals

Signal Name	Type	Description
JTAG Signals		
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.
PCH_JTAGX	I/O	This pin is used to support merged debug port topologies.
DBG_PMODE	O	ITP Power Mode Indicator. This signal is used to transmit processor and PCH power/reset information to the Debugger.
Boundry Scan Sideband Signals		
GPP_E19 / DDP1_CTRLDATA / TBT_LSX0_RXD / **BSSB_LS0_TX**	I/O	Boundary Scan Sideband Low Speed Transmit 0 for debug purposes
GPP_E18 / DDP1_CTRLCLK / TBT_LSX0_TXD / **BSSB_LS0_RX**	I/O	Boundary Scan Sideband Low Speed Receive 0 for debug purposes
GPP_E21 / DDP2_CTRLDATA / TBT_LSX1_RXD / **BSSB_LS1_TX**	I/O	Boundary Scan Sideband Low Speed Transmit 1 for debug purposes
GPP_E20 / DDP2_CTRLCLK / TBT_LSX1_TXD / **BSSB_LS1_RX**	I/O	Boundary Scan Sideband Low Speed Receive 1 for debug purposes
GPP_D10 / ISH_SPI_CLK / DDP3_CTRLDATA / TBT_LSX2_RXD / **BSSB_LS2_TX** / GSPI2_CLK	O	Boundary Scan Sideband Low Speed Transmit 2 for debug purposes
GPP_D9 / ISH_SPI_CS# / DDP3_CTRLCLK / TBT_LSX2_TXD / **BSSB_LS2_RX** / GSPI2_CS0#	I	Boundary Scan Sideband Low Speed Receive 2 for debug purposes
GPP_D12 / ISH_SPI_MOSI / DDP4_CTRLDATA / TBT_LSX3_RXD / **BSSB_LS3_TX** / GSPI2_MOSI	O	Boundary Scan Sideband Low Speed Transmit 3 for debug purposes
GPP_D11 / ISH_SPI_MISO / DDP4_CTRLCLK /	I	Boundary Scan Sideband Low Speed Receive 3 for debug purposes

continued...

Signal Name	Type	Description
TBT_LSX3_TXD / **BSSB_LS3_RX** / GSPI2_MISO		

30.3.2 I/O Signal Planes and States

Table 89. Power Planes and States for Testability Signals

Signal Name	Power Plane	Resistors	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
PCH_JTAG_TCK	Primary	Strong Internal Pull-Down	Driven Low	Driven Low	Driven Low	OFF
PCH_JTAG_TMS	Primary	Internal Pull-Up	Driven High	Driven High	Driven High	OFF
PCH_JTAG_TDI	Primary	Internal Pull-Up	Driven High	Driven High	Driven High	OFF
PCH_JTAG_TDO	Primary	External Pull-Up	Undriven	Undriven	Undriven	OFF
PCH_JTAGX[1]	Primary	Internal Strong Pull-Up (as TDO Input), Internal Strong Pull-Down (as TCK Output)	Driven High	Driven High / Driven Low	Driven High / Driven Low	OFF
DBG_PMODE	Primary	Internal Pull-Up	Driven High	Driven High	Driven High	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. Reset reference for primary well pins is RSMRST#.

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

The PCH implements three independent UART interfaces, UART0, UART1 and UART2. Each UART interface is a 4-wire interface supporting up to 6.25 Mbit/s.

The interfaces can be used in the low-speed, full-speed, and high-speed modes. The UART communicates with serial data ports that conform to the RS-232 interface protocol.

UART2 only implements the UART Host controller and does not incorporate a DMA controller which is implemented for UART0 and UART1. Therefore, UART2 is restricted to operate in PIO mode only.

The UART interfaces support the following features:

- Up to 6.25 Mbit/s Auto Flow Control mode as specified in the 16750 standard
- Transmitter Holding Register Empty (THRE) interrupt mode
- 64-byte TX and 64-byte RX host controller FIFOs
- DMA support with 64-byte DMA FIFO per channel (up to 32-byte burst)
- Functionality based on the 16550 industry standards
- Programmable character properties, such as number of data bits per character (5-8), optional parity bit (with odd or even select) and number of stop bits (1, 1.5, or 2)
- Line break generation and detection
- DMA signaling with two programmable modes
- Prioritized interrupt identification
- Programmable FIFO enable/disable
- Programmable serial data baud rate
- Modem and status lines are independently controlled
- Programmable BAUD RATE supported (baud rate = (serial clock frequency)/(16xdivisor))

NOTES

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

Table 90. Acronyms

Acronyms	Description
DMA	Direct Memory Access
UART	Universal Asynchronous Receiver/Transmitter
LSx	Low speed IO Controller

31.1 Functional Description

This section provides information on the following topics:

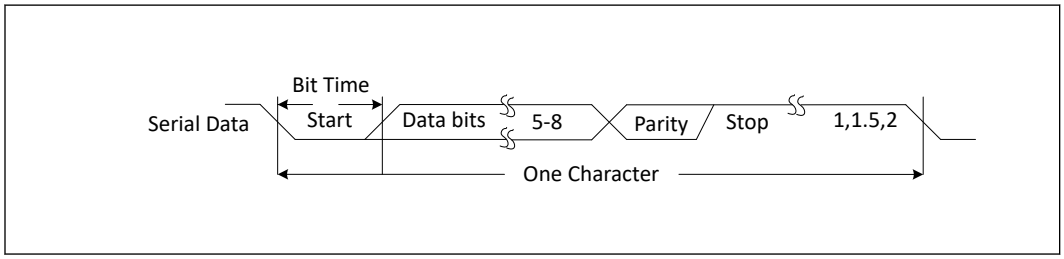
- UART Serial (RS-232) Protocols Overview
- 16550 8-bit Addressing - Debug Driver Compatibility
- DMA Controller
- Reset
- Power Management
- Interrupts
- Error Handling

31.1.1 UART Serial (RS-232) Protocols Overview

Because the serial communication between the UART host controller and the selected device is asynchronous, Start and Stop bits are used on the serial data to synchronize the two devices. The structure of serial data accompanied by Start and Stop bits is referred to as a character.

An additional parity bit may be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure to provide the UART Host Controller with the ability to perform simple error checking on the received data.

Figure 22. UART Serial Protocol



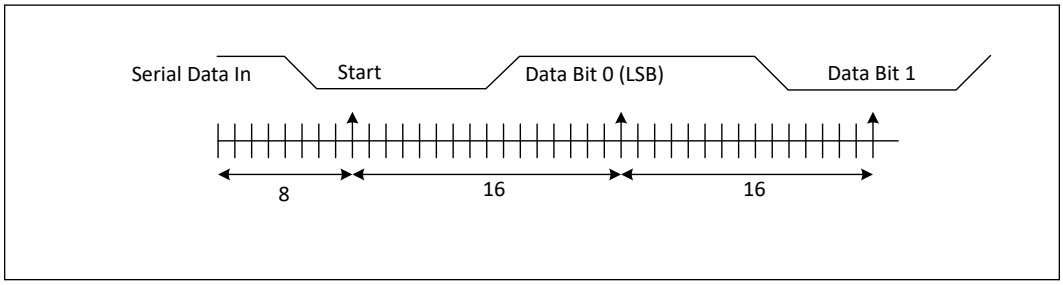
The UART Host Controller Line Control Register (LCR) is used to control the serial character characteristics. The individual bits of the data word are sent after the Start bit, starting with the least significant bit (LSB). These are followed by the optional parity bit, followed by the Stop bit(s), which 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 23. UART Receiver Serial Data Sample Points



31.1.2 16550 8-bit Addressing - Debug Driver Compatibility

NOTE

The PCH UART host controller is not compatible with legacy UART 16550 debug-port drivers. The UART host controller operates in 32-bit addressing mode only. UART 16550 legacy drivers only operate with 8-bit addressing. In order to provide compatibility with standard in-box legacy UART drivers a 16550 Legacy Driver mode has been implemented in the UART controller that will convert 8-bit addressed accesses from the 16550 legacy driver to the 32-bit addressing that the UART host controller supports.The UART 16550 8-bit Legacy mode only operates with PIO transactions. DMA transactions are not supported in this mode.

31.1.3 DMA Controller

The UART controllers 0 and 1 (UART0 and UART1) have an integrated DMA controller. Each channel contains a 64-byte FIFO. Max. burst size supported is 32 bytes.

UART controller 2 (UART2) only implements the host controllers and does not incorporate a DMA. Therefore, UART2 is restricted to operate in PIO mode only.

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

Channel Control

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

31.1.4 Reset

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

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

31.1.5 Power Management

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.

Latency Tolerance Reporting (LTR)

Latency Tolerance Reporting is used to allow the system to optimize internal power states based on dynamic data, comprehending the current platform activity and service latency requirements. The UART bus architecture, however, does not provide the architectural means to define dynamic latency tolerance messaging. Therefore, the interface supports this by reporting its service latency requirements to the platform power management controller via LTR registers.

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

1. Platform/HW Default Control. This scheme is used for usage models in which the controller's state correctly informs the platform of the current latency requirements. In this scheme, the latency requirement is a function of the controller state. The latency for transmitting data to/from its connected device at a given rate while the controller is active is representative of the active latency requirements. On the other hand if the device is not transmitting or receiving data and idle, there is no expectation for end to end latency.
2. Driver Control. This scheme is used for usage models in which the controller state does not inform the platform correctly of the current latency requirements. If the FIFOs of the connected device are much smaller than the controller FIFOs, or the connected device's end to end traffic assumptions are much smaller than the latency to restore the platform from low power state, driver control should be used.

31.1.6 Interrupts

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

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

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

31.1.7 Error Handling

Errors that might occur on the external UART signals are comprehended by the host controller and reported to the interface host controller driver through the MMIO registers.

31.2 Signal Description

Signal Name	Type	Description
GPP_H10 / **UART0_RXD** / M2_SKT2_CFG0	I	**UART 0 Receive Data**
GPP_H11 / **UART0_TXD** / M2_SKT2_CFG1	O	**UART 0 Transmit Data**
GPP_H12 / I2C7_SDA/ **UART0_RTS#** / M2_SKT2_CFG0 / ISH_GP6B / DEVSLP0B#	O	**UART 0 Request to Send**
GPP_H13 / I2C7_SCL / **UART0_CTS#** / M2_SKT2_CFG1 / ISH_GP7B / DEVSLP1B#	I	**UART 0 Clear to Send**

continued...

Signal Name	Type	Description
GPP_D17 / **UART1_RXD** / ISH_UART1_RXD	I	**UART 1 Receive Data**
GPP_D18 / **UART1_TXD** / ISH_UART1_TXD	O	**UART 1 Transmit Data**
GPP_F1 / CNV_BRI_RSP / **UART2_RXD**	I	**UART 2 Receive Data**
GPP_F2 / CNV_RGI_DT / **UART2_TXD**	O	**UART 2 Transmit Data**
GPP_F0 / CNV_BRI_DT / **UART2_RTS#**	O	**UART 2 Request to Send**
GPP_F3 / CNV_RGI_RSP / **UART2_CTS#**	I	**UART 2 Clear to Send**

31.3 Integrated Pull-Ups and Pull-Downs

None.

31.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
UART[3:0]_RXD	Primary	Undriven	Undriven	Undriven	OFF
UART[3:0]_TXD	Primary	Undriven	Undriven	Undriven	OFF
UART[3:0]_RTS#	Primary	Undriven	Undriven	Undriven	OFF
UART[3:0]_CTS#	Primary	Undriven	Undriven	Undriven	OFF
Note: 1. Reset reference for primary well pins is RSMRST#.					

31.5 LSx

LSx interface supports Four ports. Each port of the LSx controller has two bi-directional signals configured either as Tx (Output) or Rx (Input). Operating voltage of the LSx interface is 3.3 V. LSx controller is responsible for link initialization/ management of HSIO in the Thunderbolt subsystem.

31.5.1 LSx Signal Description

Signal Name	Type	Description
GPP_E19 / DDP1_CTRLDATA / **TBT_LSX0_RXD** / BSSB_LS0_TX	I	**LSx 0 Receive Data**
GPP_E18 / DDP1_CTRLCLK / **TBT_LSX0_TXD** / BSSB_LS0_RX	O	**LSx 0 Transmit Data**
		continued...

Signal Name	Type	Description
GPP_E21 / DDP2_CTRLDATA / **TBT_LSX1_RXD** / BSSB_LS1_TX	I	**LSx 1 Receive Data**
GPP_E20 / DDP2_CTRLCLK / **TBT_LSX1_TXD** / BSSB_LS1_RX	O	**LSx 1 Transmit Data**
GPP_D10 / ISH_SPI_CLK / DDP3_CTRLDATA / **TBT_LSX2_RXD** / BSSB_LS2_TX / GSPI2_CLKI	I	**LSx 2 Receive Data**
GPP_D9 / ISH_SPI_CS# / DDP3_CTRLCLK / **TBT_LSX2_TXD** / BSSB_LS2_RX / GSPI2_CS0#	O	**LSx 2 Transmit Data**
GPP_D12 / ISH_SPI_MOSI / DDP4_CTRLDATA / **TBT_LSX3_RXD** / BSSB_LS3_TX / GSPI2_MOSI	I	**LSx 3 Receive Data**
GPP_D11 / ISH_SPI_MISO / DDP4_CTRLCLK / **TBT_LSX3_TXD** / BSSB_LS3_RX / GSPI2_MISO	O	**LSx 3 Transmit Data**

31.5.2 Integrated Pull-Ups and Pull-Downs

None.

31.5.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
TBT_LSX[0:3]_RXD	Primary	Undriven	Undriven	Undriven	OFF
TBT_LSX[0:3]_TXD	Primary	Undriven	Undriven	Undriven	OFF
Note: 1. Reset reference for primary well pins is RSMRST#.					

32.0 Universal Serial Bus (USB)

The PCH implements an xHCI USB 3.2 controller which provides support for up to 10 USB 2.0 signal pairs and 4 USB 3.2 signal pairs. The xHCI controller supports wake up from sleep states S1-S4. The xHCI controller supports up to 64 devices for a maximum number of 2048 Asynchronous endpoints (Control / Bulk) or maximum number of 128 Periodic endpoints (Interrupt / isochronous).

Each walk-up USB 3.2 capable port must includes USB 3.2 and USB 2.0 signaling.

Table 91. Acronyms

Acronyms	Description
xHCI	eXtensible Host Controller Interface

Table 92. References

Specification	Location
USB 4.0 Specification	www.usb.org
USB 3.2 Specification	
USB 3.1 Specification	
USB 3.0 Specification	
USB 2.0 Specification	

32.1 Functional Description

This section provides information on the following topics:

1. eXtensible Host Controller Interface (xHCI) Controller
2. USB Dual Role Support - eXtensible Device Controller Interface (xDCI) Controller

32.1.1 eXtensible Host Controller Interface (xHCI) Controller

The eXtensible Host Controller Interface (xHCI) allows data transfer speed up to 10 Gb/s for USB 3.2 Gen 2x1 ports, and 5 Gb/s for USB 3.2 Gen 1x1 ports. The xHCI supports SuperSpeed USB 10 Gbps, SuperSpeed USB 5 Gbps, High-Speed (HS), Full-Speed (FS) and Low-Speed (LS) traffic on the bus. The xHCI supports USB Debug port on all the USB ports. The xHCI also supports USB Attached SCIS Protocol (UASP).

32.1.2 USB Dual Role Support - eXtensible Device Controller Interface (xDCI) Controller

The USB subsystem also supports Dual Role Capability. The xHCI is paired with a standalone eXtensible Device Controller Interface (xDCI) to provide dual role functionality. The USB subsystem incorporates a xDCI USB 3.2 Gen 1x1 (5 Gb/s) device controller. The dual role capability splits the support for SuperSpeed USB 5

Gbps on the CPU xDCI controller, and High-Speed (HS) on the PCH xDCI controller. These device controllers are instantiated as a separate PCI function. The USB implementation is compliant to the Device specification and supports host/device only through the USB Type-C* connector.

The xDCI shares all USB ports with the host controller, with the ownership of the port being decided based the USB Power Delivery specification. Since all the ports support device mode, xDCI enabling must be extended by System BIOS and EC. While the port is mapped to the device controller, the host controller Rx detection must always indicate a disconnected port. Only one port can be connected (and active) to the device controller at one time. Any subsequently connection will not be established.

32.1.3 AUX BIAS Control - USB Type-C Implementation with no Retimer

On processor, which support integrated USB Type-C[1] subsystem, the AUX BIAS control is required on the USB Type-C implementation (without retimer) for orientation connections.

The functionality is muxed with certain GPIO pins. Refer to the GPIO implementation document for more information on the muxing and supported GPIO pin on the specific platform. In order to use the GPIO pin correctly for AUX BIAS control, the correct native functionality need to be configured and the correct Virtual Wire Index bit position need to be programmed in the BIOS policy.

Figure 24. GPIO - Virtual Wire Index Bit Mapping

GPIO Pin Group	Virtual Wire Index	Bit Position*
USB_C_GPP_[A7:A0]	10h	[7h:0h]
USB_C_GPP_[A15:A8]	11h	[7h:0h]
USB_C_GPP_[A23:A16]	12h	[7h:0h]
USB_C_GPP_[B7:B0]	13h	[7h:0h]
USB_C_GPP_[B15:B8]	14h	[7h:0h]
USB_C_GPP_[B23:B16]	15h	[7h:0h]
USB_C_GPP_[C7:C0]	13h	[7h:0h]
USB_C_GPP_[C15:C8]	14h	[7h:0h]
USB_C_GPP_[C23:C16]	15h	[3h:0h]
USB_C_GPP_[D7:D0]	10h	[7h:0h]
USB_C_GPP_[D15:D8]	11h	[7h:0h]
USB_C_GPP_[D19:D16]	12h	[3h:0h]
USB_C_GPP_[E7:E0]	16h	[7h:0h]
USB_C_GPP_[E15:E8]	17h	[7h:0h]
USB_C_GPP_[E23:E16]	18h	[7h:0h]
USB_C_GPP_[F7:F0]	10h	[7h:0h]
USB_C_GPP_[F15:F8]	11h	[7h:0h]
USB_C_GPP_[F23:F16]	12h	[7h:0h]
USB_C_GPP_[H3:H0]	12h	[7h:4h]
USB_C_GPP_[H11:H4]	13h	[7h:0h]
USB_C_GPP_[H19:H12]	14h	[7h:0h]
USB_C_GPP_[H23:H20]	15h	[3h:0h]

NOTE

1. The bit position corresponds to each corresponding GPIO pin in the group.
   Example: the bit position for USB_C_GPP_A0 is bit 0h in Virtual Wire Index 10h.

32.2 Signal Description

Name	Type	Description
PCIE1_RXN/**USB32_1_RXN** PCIE1_RXP/**USB32_1_RXP**	I	**USB 3.2 Differential Receive Pair 1**: These are USB 3.2-based high-speed differential signals for Port 1. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
PCIE1_TXN/**USB32_1_TXN** PCIE1_TXP/**USB32_1_TXP**	O	**USB 3.2 Differential Transmit Pair 1**: These are USB 3.2-based high-speed differential signals for Port 1. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
PCIE2_RXN/**USB32_2_RXN** PCIE2_RXP/**USB32_2_RXP**	I	**USB 3.2 Differential Receive Pair 2**: These are USB 3.2-based high-speed differential signals for Port 2. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
PCIE2_TXN/**USB32_2_TXN** PCIE2_TXP/**USB32_2_TXP**	O	**USB 3.2 Differential Transmit Pair 2**: These are USB 3.2-based high-speed differential signals for Port 2. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
PCIE3_RXN/**USB32_3_RXN** PCIE3_RXP/**USB32_3_RXP**	I	**USB 3.2 Differential Receive Pair 3**: These are USB 3.2-based high-speed differential signals for Port 3. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
PCIE3_TXN/**USB32_3_TXN** PCIE3_TXP/**USB32_3_TXP**	O	**USB 3.2 Differential Transmit Pair 3**: These are USB 3.2-based high-speed differential signals for Port 3. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
PCIE4_RXN/**USB32_4_RXN** PCIE4_RXP/**USB32_4_RXP**	I	**USB 3.2 Differential Receive Pair 4**: These are USB 3.2-based high-speed differential signals for Port 4. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
PCIE4_TXN/**USB32_4_TXN** PCIE4_TXP/**USB32_4_TXP**	O	**USB 3.2 Differential Transmit Pair 4**: These are USB 3.2-based high-speed differential signals for Port 4. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB2P_1/ **USB2N_1**	I/O	**USB 2.0 Port 1 Transmit/Receive Differential Pair 1:** This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB2P_2/ **USB2N_2**	I/O	**USB 2.0 Port 2 Transmit/Receive Differential Pair 2:** This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB2P_3/ **USB2N_3**	I/O	**USB 2.0 Port 3Transmit/Receive Differential Pair 3:** This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB2P_4/ **USB2N_4**	I/O	**USB 2.0 Port 4 Transmit/Receive Differential Pair 4:** This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB2P_5/ **USB2N_5**	I/O	**USB 2.0 Port 5 Transmit/Receive Differential Pair 5:** This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB2P_6/ **USB2N_6**	I/O	**USB 2.0 Port 6 Transmit/Receive Differential Pair 6:** This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.

continued...

Name	Type	Description
USB2P_7/ USB2N_7	I/O	**USB 2.0 Port 7 Transmit/Receive Differential Pair 7:** This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB2P_8/ USB2N_8	I/O	**USB 2.0 Port 8 Transmit/Receive Differential Pair 8:** This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB2P_9/ USB2N_9	I/O	**USB 2.0 Port 9 Transmit/Receive Differential Pair 9:** This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB2P_10/ USB2N_10	I/O	**USB 2.0 Port 10 Transmit/Receive Differential Pair 10:** This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
GPP_E9/**USB_OC0#**/ISH_GP4	I	**Overcurrent Indicators**: This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred. When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required. *Notes:* 1. OC# pins are not 5 V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..
GPP_A14/**USB_OC1#**/DDSP_HPD3/ DISP_MISC3	I	**Overcurrent Indicators**: This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred. When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required. *Notes:* 1. OC# pins are not 5 V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..
GPP_A15/**USB_OC2#**/DDSP_HPD4/ DISP_MISC4	I	**Overcurrent Indicators**: This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred. When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required. *Notes:* 1. OC# pins are not 5 V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..
GPP_A16 / **USB_OC3#**/ISH_GP5	I	**Overcurrent Indicators**: This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred. When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required. *Notes:* 1. OC# pins are not 5 V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..
USB_ID	I	ID detect for USB Device mode.

continued...

Name	Type	Description
		Note: This HW signal is not used on the PCH for dual role mode selection. The switching of USB port role is done through message from EC/PD over SMLink1. This signal should be connected to ground.
USB2_COMP	I	USB Resistor Bias, analog connection points for an external resistor Ω ± 1% connected to GND.
USB_VBUSSENSE	I	VBUS Sense for USB Device mode. This HW signal is not used on the PCH for USB device mode functionality. This signal should be connected to ground.

32.3 Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value	Notes
USB2P_[10:1]	Internal Pull-down	14.25–24.8 kΩ	1
USB2N_[10:1]	Internal Pull-down	14.25–24.8 kΩ	1
USB_ID	Internal Weak Pull-up	14.25–24.8 kΩ	If this signal is not in use, then the pin shall be connected directly to ground.
Note: 1. Series resistors (45 Ω ±10%)			

32.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[2]	Immediately After Reset[2]	S4/S5	Deep Sx
USB32_[4:1]_RXN **USB32_[4:1]_RXP**	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
USB32_[4:1]_TXN **USB32_[4:1]_TXP**	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
USB2N_[10:1]	DSW	Internal Pull-down	Internal Pull-down	Internal Pull-down	Internal Pull-down
USB2P_[10:1]	DSW	Internal Pull-down	Internal Pull-down	Internal Pull-down	Internal Pull-down
USB_OC0#	Primary	Undriven	Undriven	Undriven	OFF
USB_OC1#	Primary	Undriven	Undriven	Undriven	OFF
USB_OC2#	Primary	Undriven	Undriven	Undriven	OFF
USB_OC3#	Primary	Undriven	Undriven	Undriven	OFF
USB_VBUSSENSE	Primary	Undriven	Undriven	Undriven	OFF
USB_ID[1]	Primary	Internal Pull-up	Undriven/Internal Pull-up	Undriven/Internal Pull-up	OFF
USB2_COMP	Primary	Undriven	Undriven	Undriven	OFF
Notes: 1. The USB_ID pin is pulled-up internally. 2. Reset reference for primary well pins is RSMRST# and DSW well pins is DSW_PWROK.					

32.5 Supported USB 2.0 Ports

Due to the USB 2.0 port requirement for integrated Bluetooth* functionality with the integrated Intel® Wireless-AX (CNVi) solution, the following USB port will be available:

The following table describes the number of USB 2.0 ports supported :

Figure 25. Supported USB 2.0 Ports on the PCH

CHIPSET SKU	Max USB 2.0 Nbr of Ports	USB 2.0 P1	USB 2.0 P2	USB 2.0 P3	USB 2.0 P4	USB 2.0 P5	USB 2.0 P6	USB 2.0 P7	USB 2.0 P8	USB 2.0 P9	USB 2.0 P10 (or Intel® Wireless-AX)	USBr1	USBr2
PREMIUM P	10												

Port Enabled

33.0 Connectivity Integrated (CNVi)

Connectivity Integrated (CNVi) is a general term referring to a family of connectivity solutions which are based on the Connectivity Controller family. The common component of all these solutions is the Connectivity Controller IP, which is a hard macro embedded in various Intel SoC chips.

The Integrated Connectivity (CNVi) solution consists of the following entities:

- The containing chip (SoC or PCH which contains the Connectivity Controller IP)
- Buttress (as applicable to each platform, and coupled the Connectivity Controller IP)
- Companion RF chip that is in a pre-certified module (i.e., M.2-2230, M.2-1216) or soldered as chip on board.

The main blocks of the integrated Connectivity solution are partitioned according to the following:

Table 93. Acronyms

Acronyms	Description
BRI	Bluetooth* Radio Interface
CNVi	Connectivity Integrated
PCH	Platform Controller Hub
RGI	Radio Generic interface
SoC	System On Chip
IP	Literally, Intellectual Property. IP refers to architecture, design, validation, and software components collectively delivered to enable one or more specific SoC features
MFUART	Multifunction Universal Asynchronous Receiver/Transmitter
UART	Universal Asynchronous Receiver/Transmitter

Table 94. References

Specification	Location
M.2 Specification	https://pcisig.com/specifications/pciexpress/M.2_Specification/
MIPI* Alliance specification for D-PHY v1.2	http://www.mipi.org/specifications

33.1 Functional Description

The main blocks of the integrated Connectivity solution are partitioned according to the following:

- **Connectivity Controller IP** contains:
  - Interfaces to the PCH
  - Debug and testing interfaces

  - Power management and clock Interfaces
  - Interface to the Companion RF module (CRF)
  - Interface to physical I/O pins controlled by the PCH
  - Interfaces to the LTE modem via PCH GPIO
- **Companion RF (CRF)**: This is the integrated connectivity M.2 module. The CRF Top contains:
  - Debug and testing interfaces
  - Power and clock Interfaces
  - Interface to the Connectivity Controller chip
- **Physical I/O Pins**: The SCU units are responsible for generating and controlling the power and clock resources of Connectivity Controller and CRF. There are unique SCUs in Connectivity Controller and CRF and their operation is coordinated due to power and clock dependencies. This coordination is achieved by signaling over a control bus (AUX) connecting Connectivity Controller and CRF.

Both Connectivity Controller and CRF have a dedicated AUX bus and arbiter. These two AUX buses are connected by a special interface that connects over the RGI bus. Each of the Connectivity Controller and CRF cores is dedicated to handle a specific connectivity function (Wi-Fi, Bluetooth).

Only the digital part of the connectivity function is located in Connectivity Controller cores, while the CRF cores handle some digital, but mostly analog and RF functionality. Each core in the Connectivity Controller has an interface to the host and an interface to its counterpart in CRF. CRF cores include an analog part which is connected to board level RF circuitry and to an antenna.

33.2 Signal Description

Signal Name	Type	Description
GPIO fixed functions (Signals for Integrated Connectivity (CNVi) and Discrete Connectivity (CNVd) functions		
GPP_R4 / HDA_RST# / **I2S2_SCLK** / DMIC_CLKA0	I/O	For CNVi: Unused For discrete connectivity with UART host support: Optional Bluetooth* I^2S bus clock
GPP_F4 / **CNV_RF_RESET#**	I/O	For CNVi: RF companion (CRF) reset signal, active low. Require a 75 kohm Pull-Down on platform/motherboard level. It is recommended not to use it for bootstrapping during early Platform init flows. For discrete connectivity with UART host support: Optional Bluetooth* I^2S bus sync
GPP_R6 / **I2S2_TXD** / DMIC_CLK_A1	O	For discrete connectivity with UART host Bluetooth* support: Optional Bluetooth* I^2S bus data output (input to Bluetooth* module)
GPP_R7 / **I2S2_RXD** / DMIC_DATA1	I	For CNVi: Unused. For discrete connectivity with UART host support: Optional Bluetooth* I^2S bus data output (input to Bluetooth* module)
GPP_F0 / **CNV_BRI_DT** / UART2_RTS#	O	For CNVi: BRI bus TX. For discrete connectivity with UART host support: Bluetooth* UART RTS#
GPP_F1 / **CNV_BRI_RSP** / UART2_RXD	I	For CNVi: BRI bus RX. For discrete connectivity with UART host support: Bluetooth* UART RXD

continued...

Signal Name	Type	Description
GPP_F2 / **CNV_RGI_DT** / UART2_TXD	O	For CNVi: RGI bus TX. RGI_DT is used by the platform to strap presence of the CRF. Requires weak pull up of 20Kohm on the platform. For discrete connectivity with UART host support: Bluetooth* UART TXD
GPP_F3 / **CNV_RGI_RSP** / UART2_CTS#	I	For CNVi: RGI bus RX. For discrete connectivity with UART host support: Bluetooth* UART CTS#
GPP_F5 / MODEM_CLKREQ / **CRF_XTAL_CLKREQ**	O	For CNVi: SOC to Companion RF (CRF) clock request indication : A dedicated crystal clock request signal from CNVi to Companion RF. See also MODEM_CLKREQ. This clock request will only bring up the CRF crystal clock without bringing up the PCH crystal clock. Note: MODEM_CLKREQ or CRF_XTAL_CLKREQ function can be selected for CRF clock request with different impact but to the PCH crystal clock. In non-XTAL shared platforms, CRF_XTAL_CLKREQ is recommended.
GPP_F5 / **MODEM_CLKREQ** / CRF_XTAL_CLKREQ	O	For CNVi: SOC to Companion RF (CRF) clock request indication : A crystal clock request from PCH to the CNVi Companion RF. See also CRF_XTAL_CLKREQ. This clock request will bring up the PCH crystal clock together with the CRF crystal clock. Note: MODEM_CLKREQ or CRF_XTAL_CLKREQ function can be selected for CRF clock request with different impact but to the PCH crystal clock.
GPP_F6 / **CNV_PA_BLANKING**	I/O	For CNVi and discrete connectivity : Optional WLAN/Bluetooth* WWAN co-existence signal. Used to be co-existence signal for external GNSS solution
GPP_H8 / I2C4_SDA / **CNV_MFUART2_RXD**	I	For CNVi and discrete connectivity: Optional WLAN/Bluetooth* WWAN co-existence signal (Input)
GPP_H9 / I2C4_SCL / **CNV_MFUART2_TXD**	O	For CNVi and discrete connectivity : Optional WLAN/Bluetooth* WWAN co-existence signal (Output)
Fixed special purpose I/O		
CNV_WT_CLKP	O	CNVio bus TX CLK+
CNV_WT_CLKN	O	CNVio bus TX CLK-
CNV_WT_D0P	O	CNVio bus Lane 0 TX+
CNV_WT_D0N	O	CNVio bus Lane 0 TX-
CNV_WT_D1P	O	CNVio bus Lane 1 TX+
CNV_WT_D1N	O	CNVio bus Lane 1 TX-
CNV_WR_CLKP	I	CNVio bus RX CLK+
CNV_WR_CLKN	I	CNVio bus RX CLK-
CNV_WR_D0P	I	CNVio bus Lane 0 RX+
CNV_WR_D0N	I	CNVio bus Lane 0 RX-
CNV_WR_D1P	I	CNVio bus Lane 1 RX+
CNV_WR_D1N	I	CNVio bus Lane 1 RX-
Selectable special purpose I/O		

continued...

Signal Name	Type	Description
USB2P_10	I/O	Bluetooth* USB host bus (positive) for discrete connectivity. Optional to connect to a Bluetooth* USB+ pin on the Bluetooth* module. is the recommended port but other USB 2.0 ports can be selected for this function.
USB2N_10	I/O	Bluetooth* USB host bus (negative) for discrete connectivity. Optional to connect to a Bluetooth* USB+ pin on the Bluetooth* module. is the recommended port but other USB 2.0 ports can be selected for this function.
PCIE12_TXP	O	Wi-Fi* PCIe* host bus TX (positive) for discrete connectivity. Optional to connect to a Wi-Fi* PCIe* PERp0 pin on the Wi-Fi* module. This is the recommended port but other PCIe* ports can be selected for this function.
PCIE12_TXN	O	Wi-Fi* PCIe* host bus TX (negative) for discrete connectivity. Optional to connect to a Wi-Fi* PCIe* PERn0 pin on the Wi-Fi* module. This is the recommended port but other PCIe* ports can be selected for this function.
PCIE12_RXP	I	Wi-Fi* PCIe* host bus RX (positive) for discrete connectivity. Optional to connect to a Wi-Fi* PCIe* PETp0 pin on the Wi-Fi* module. This is the recommended port but other PCIe* ports can be selected for this function.
PCIE12_RXN	I	Wi-Fi* PCIe* host bus RX (negative) for discrete connectivity. Optional to connect to a Wi-Fi* PCIe* PETn0 pin on the Wi-Fi* module. This is the recommended port but other PCIe* ports can be selected for this function.
CLKOUT_PCIE_P3	O	Wi-Fi* PCIe* host bus clock (positive) for discrete connectivity. Optional to connect to a Wi-Fi* PCIe* REFCLKp pin on the Wi-Fi* module. This is the recommended clock signal but other PCIe* clocks can be selected for this function.
CLKOUT_PCIE_N3	O	Wi-Fi* PCIe* host bus clock (negative) for discrete connectivity. Optional to connect to a Wi-Fi* PCIe* REFCLKp pin on the Wi-Fi* module. This is the recommended clock signal but other PCIe* clocks can be selected for this function.
CL_RST#	O	Wi-Fi* CLINK host bus reset for discrete connectivity with CLINK support (Intel® vPro™). Optional to connect to a Wi-Fi* CLINK reset pin on the Intel® vPro™ Wi-Fi* module.
CL_DATA	I/O	Wi-Fi* CLINK host bus data for discrete connectivity with CLINK support (Intel® vPro™). Optional to connect to a Wi-Fi* CLINK data pin on the Intel® vPro™ Wi-Fi* module.
CL_CLK	O	Wi-Fi* CLINK host bus clock for discrete connectivity with CLINK support (Intel® vPro™). Optional to connect to a Wi-Fi* CLINK clock pin on the Intel® vPro™ Wi-Fi* module.
W_Disable1# (GPIO)	O	Used for Wi-Fi* RF-Kill control. This pin can be connected to a platform switch or to SoC GPIOs (recommendation- if possible do not use GPIOs that have Platform impact as "bootstraps" during platform init). The signal must keep value in Sx state (configured in BIOS) The W_Disable signal have a Pull-up embedded in the CRF silicon, (this is an Active-Low signal).
W_Disable2# (GPIO)	O	Used for Bluetooth* RF-Kill control. This pin can be connected to a platform switch or to SoC GPIOs (recommendation- if possible do not use GPIOs that have Platform impact as "bootstraps" during platform init). The signal must keep value in Sx state (configured in BIOS) The W_Disable signal have a Pull-up embedded in the CRF silicon, (this is an Active-Low signal).

33.3 Integrated Pull-ups and Pull-downs

Signal Name	Resistor	Value	Notes
CNV_BRI_RSP	Pull up	20 kohm	
CNV_RGI_RSP	Pull up	20 kohm	

33.4 I/O Signal Planes and States

Signal Name	Power plane	During Reset[1]	Immediately After Reset[1]	S4/S5	Deep Sx
CNV_RF_RESET#	Primary	Driven	Driven	Driven	OFF
CNV_MFUART2_RXD	Primary	Undriven	Undriven	Undriven	OFF
CNV_MFUART2_TXD	Primary	Undriven	Undriven	Undriven	OFF
CNV_BRI_DT	Primary	Driven	Driven	Driven	OFF
CNV_BRI_RSP	Primary	Powered (input, PU)	Powered (input, PU)	Powered (input, PU)	OFF
CNV_RGI_DT	Primary	Driven	Driven	Driven	OFF
CNV_RGI_RSP	Primary	Powered (input, PU)	Powered (input, PU)	Powered (input, PU)	OFF
CNV_WT_CLKP	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_CLKN	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_D0P	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_D0N	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_D1P	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_D1N	Primary	Undriven	Undriven	Driven	OFF
CNV_WR_CLKP	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_CLKN	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_D0P	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_D0N	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_D1P	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_D1N	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WT_RCOMP	Primary	Undriven	Undriven	Driven	OFF

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

34.0 GPIO Serial Expander

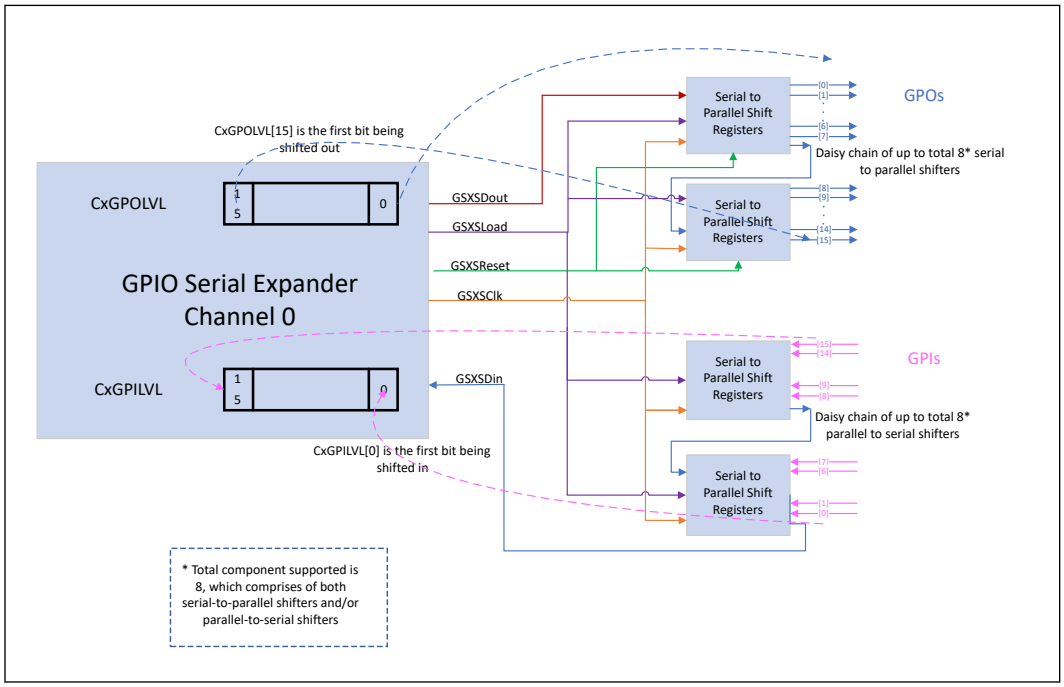
GPIO Serial Expander (GSX) is the capability provided by the PCH to expand the GPIOs on a platform that needs more GPIOs than the ones provided by the PCH. The solution requires external shift register discrete components.

34.1 Functional Description

GPIO Serial Expander (GSX) uses serial-to-parallel or parallel-to-serial shift register discrete components to increase number of the GPIO pins for system use. It expands in the multiples of 8 for input or output with 8 pins per expander. The total shift register component supported is 8, which can expand the GPIOs by up to 64.

The below figure illustrates a GPIO expansion topology with 16 GPIs and 16 GPOs.

Figure 26. Example of GSX Topology



Coming out of system reset, GSX is in reset with the following behaviors:

- GSXSRESET# asserted by default. The signal remains asserted until BIOS/SW initialization has been completed and CxCMD.ST set to 1.
- GSXSLOAD is 0 by default until CxCMD.ST is set to 1.
- GSXSCLK is not toggling until CxCMD.ST is set to 1.

34.2 Signal Description

Signal Name	Type	Description
GPP_F12 / **GSXDOUT** / THC1_SPI2_IO0 / GSPI1_MOSI / I2C1A_SCL	O	GPIO Serial Expander Controller Data Out
GPP_F13 / **GSXSLOAD** / THC1_SPI2_IO1 / GSPI1_MISIO / I2C1A_SDA	O	GPIO Serial Expander Controller Serial Load
GPP_F14 / **GSXDIN** / THC1_SPI2_IO2	I	GPIO Serial Expander Controller Data In
GPP_F15 / **GSXSRESET#** / THC1_SPI2_IO3	O	GPIO Serial Expander Controller Serial Reset
GPP_F16 / **GSXCLK** / THC1_SPI2_CS# / GSP1_CS0#	O	GPIO Serial Expander Controller Clock

34.3 Integrated Pull-ups and Pull-downs

None

35.0 Universal Flash Storage (UFS)

The Universal Flash System (UFS) is the next generation storage standard that is to replace eMMC. UFS includes the feature set of eMMC as a subset and defines a unique feature set that provides low power consumption, high data throughput, low electromagnetic interference and optimization for mass memory subsystem efficiency.

PCH supports for UFS version 2.1 specification.

NOTE

UFS is supported on P 282 platforms. UFS is not supported on P 682 platforms.

35.1 Functional Description

UFS adopts the latest technology from other industrial standards,

- MIPI M-PHY specification version 3.1 for physical layer
- MIPI UniPro specification version 1.61 for interconnect layer
- INCTS T10 SCSI standards (SBC, SPC and SAM) for command sets and architecture model

The UFS Host Controller handles UFS Protocol at transmission level, packing data, adding cyclic redundancy check (CRC), start/end bit, and checking for transaction format correctness.

35.2 Signals Description

Signal Name	Type	Description
PCIE9_TXP / **UFS10_TXP**	O	UFS port 1 lane 0 transmit signal
PCIE9_TXN / **UFS10_TXN**	O	UFS port 1 lane 0 transmit signal
PCIE9_RXP / **UFS10_RXP**	I	UFS port 1 lane 0 receive signal
PCIE9_RXN / **UFS10_RXN**	I	UFS port 1 lane 0 receive signal
PCIE10_TXP / **UFS11_TXP**	O	UFS port 1 lane 1 transmit signal
PCIE10_TXN / **UFS11_TXN**	O	UFS port 1 lane 1 transmit signal
PCIE10_RXP / **UFS11_RXP**	I	UFS port 1 lane 1 receive signal
PCIE10_RXN / **UFS11_RXN**	I	UFS port 1 lane 1 receive signal
CLKOUT_PCIE_N4 / **UFS_REF_CLK**	O	UFS reference clock signal (19.2 MHz).

continued...

Signal Name	Type	Description
		Note: Level shifter is required for this signal to meet UFS specification.
UFS_RESET#	O	Unconnected pin (UFS device reset should be connected to a level shifted version of platform reset signal).

35.3 I/O Signals Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S4/S5	Deep Sx
UFS1x_TXP/N, UFS1x_RXP/N	Primary	Driven Low	Driven Low	Driven Low	OFF
UFS_RESET#	Primary	Driven Low	Driven High	Driven Low	OFF
UFS_REF_CLK	Primary	Undriven	Undriven	Undriven	Undriven

36.0 Private Configuration Space Target Port ID

The PCH incorporates a wide variety of devices and functions. The registers within these devices are mainly accessed through the primary interface, such as PCI configuration space and IO/MMIO space. Some devices also have registers that are distributed within the PCH Private Configuration Space at individual endpoints (Target Port IDs) which are only accessible through the PCH Sideband Interface. These PCH Private Configuration Space Registers can be addressed via SBREG_BAR or through SBI Index Data pair programming.

Table 95. Private Configuration Space Register Target Port IDs

PCH Device/Function Type	Target Port ID
FIA Configuration	CFh
General Purpose I/O (GPIO) Community 0	6Eh
General Purpose I/O (GPIO) Community 1	6Dh
General Purpose I/O (GPIO) Community 2	6Ch
General Purpose I/O (GPIO) Community 4	6Ah
DCI	B8h
PCIe Controller #1 (SPA)	80h
PCIe Controller #2 (SPB)	81h
PCIe Controller #3 (SPC)	82h
SATA	D9h
SMBus	C6h
eSPI / SPI	72h
xHCI	70h
CNVi	73h
HSIO Strap Configuration	89h
Real Time Clock (RTC)	C3h
Processor Interface, 8254 Timer, HPET, APIC	C4h
USB 2.0	CAh
UART, I^2C, GSPI	CBh
Integrated Clock Controller (ICC)	DCh
CSI-2 Interface	A1h
General Purpose I/O (GPIO) Community 5	69h
USB Dual Role / OTG	E5h
MODPHY0	ABh

continued...

PCH Device/Function Type	Target Port ID
MODPHY1	AAh
MODPHY2	A9h
MODPHY3	A8h
Intel® Trace Hub	B6h

37.0 Miscellaneous Signals

37.1 Signal Description

Table 96. Signal Descriptions

Name	Type	Description
GPP_D0 / ISH_GP0 / **BK0** / SBK0	OD	**Blink BK 0:** This function provides the blink (or PWM) capability. The blink/PWM frequency and duty cycle is programmable through the PWM Control register. Refer to Volume 2 for details.
GPP_D1 / ISH_GP1 / **BK1** / SBK1	OD	**Blink BK 1:** This function provides the blink (or PWM) capability. The blink/PWM frequency and duty cycle is programmable through the PWM Control register. Refer to Volume 2 for details.
GPP_D2 / ISH_GP2 / **BK2** / SBK2	OD	**Blink BK 2:** This function provides the blink (or PWM) capability. The blink/PWM frequency and duty cycle is programmable through the PWM Control register. Refer to Volume 2 for details.
GPP_D3/ISH_GP3/**BK3**/SBK3	OD	**Blink BK 3:** This function provides the blink (or PWM) capability. The blink/PWM frequency and duty cycle is programmable through the PWM Control register. Refer to Volume 2 for details.
GPP_D4 / IMGCLKOUT0 / **BK4** / SBK4	OD	**Blink BK 4:** This function provides the blink (or PWM) capability. The blink/PWM frequency and duty cycle is programmable through the PWM Control register. Refer to Volume 2 for details.
GPP_E22 / DDPA_CTRLCLK / **DNX_FORCE_RELOAD**	I	**Download and Execute (DnX):**Intel® CSME ROM samples this pin any time ROM begins execution. This includes the following conditions: • G3 Exit. • Sx, Moff Exit. • Cold Reset(Host Reset with Power Cycle) Exit. • Warm Reset(Host Reset without Power Cycle) Exit if Intel® CSME was shutdown in Warm Reset. • 0 => No DnX; 1=> Enter DnX mode *Note:* This pin must not be sampled high at the sampling time for normal operation.
GPP_E0 / **SATAXPCIE0** / SATAGP0	I	**SATA port 0 or PCIe port mux select** : This is used to select SATA/PCIe function to support implementations like SATA Express or mSATA.
GPP_A12 / **SATAXPCIE1** / SATAGP1	I	**SATA port 1 or PCIe port mux select** : This is used to select SATA/PCIe function to support implementations like SATA Express or mSATA.
GPP_D0 / ISH_GP0 / BK0 / **SBK0**	OD	**Serial Blink SBK 0:** This function provides the capability to serialize POST or other messages on the pin to a serial monitor. The Serial Blink message is programmed through the Serial Blink Command/Status and Serial Blink Data registers. Refer to Volume 2 for details.
GPP_D1 / ISH_GP1 / BK1 / **SBK1**	OD	**Serial Blink SBK 1:** This function provides the capability to serialize POST or other messages on the pin to a serial monitor. The Serial Blink message is programmed through the Serial Blink Command/Status and Serial Blink Data registers. Refer to Volume 2 for details.
GPP_D2 / ISH_GP2 / BK2 / **SBK2**	OD	**Serial Blink SBK 2:** This function provides the capability to serialize POST or other messages on the pin to a serial monitor. The Serial Blink message is programmed through the Serial Blink Command/Status and Serial Blink Data registers. Refer to Volume 2 for details.

continued...

Name	Type	Description
GPP_D3/ISH_GP3/BK3/**SBK3**	OD	**Serial Blink SBK 3:** This function provides the capability to serialize POST or other messages on the pin to a serial monitor. The Serial Blink message is programmed through the Serial Blink Command/Status and Serial Blink Data registers. Refer to Volume 2 for details.
GPP_D4 / IMGCLKOUT0 / BK4 / **SBK4**	OD	**Serial Blink SBK 4:** This function provides the capability to serialize POST or other messages on the pin to a serial monitor. The Serial Blink message is programmed through the Serial Blink Command/Status and Serial Blink Data registers. Refer to Volume 2 for details.
GPP_B15 / **TIME_SYNC0** / ISH_GP7	I	**Time Synchronization GPIO 0**: Timed GPIO event for time synchronization for interfaces that do not support time synchronization natively.
GPP_F9 / **BOOTMPC**	O	**Boot Media Power Control**: If BOOTMPC is 0 - Powered Off. If BOOTMPC is 1 - Powered On.
GPP_H10/UART0_RXD/ **M2_SKT2_CFG0**	I	**M.2 Socket 2 Configuration :** This is used to select module type and main host interface. Refer to 'Socket 2 Module Configuration' table in the PCI-SIG M.2 Specification.
GPP_H11/UART0_TXD/ **M2_SKT2_CFG1**	I	**M.2 Socket 2 Configuration :** This is used to select module type and main host interface. Refer to 'Socket 2 Module Configuration' table in the PCI-SIG M.2 Specification.
GPP_H12/I2C7_SDA/ UART0_RTS#/**M2_SKT2_CFG2**/ ISH_GP6B/DEVSLP0B	I	**M.2 Socket 2 Configuration :** This is used to select module type and main host interface. Refer to 'Socket 2 Module Configuration' table in the PCI-SIG M.2 Specification.
GPP_H13/I2C7_SCL/ UART0_CTS#/**M2_SKT2_CFG3**/ ISH_GP7B/DEVSLP1B	I	**M.2 Socket 2 Configuration :** This is used to select module type and main host interface. Refer to 'Socket 2 Module Configuration' table in the PCI-SIG M.2 Specification.
MPHY_RCOMPP	I	100 ohm (+/- 1%) connected between MPHY_RCOMPP and MPHY_RCOMPN
MPHY_RCOMPN	I	100 ohm (+/- 1%) connected between MPHY_RCOMPP and MPHY_RCOMPN
DMI_RCOMP	I	DMI Rcomp Signal (50 ohm +/- 1% pulldown to ground)

37.2 Integrated Pull-Ups and Pull-Downs

Table 97. Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value
SATAXPCIE0	Pull-down	20 kohm
SATAXPCIE1	Pull-down	20 kohm

37.3 I/O Signal Planes and States

None