

Intel® 800 Series Chipset Family Platform Controller Hub (PCH)

Datasheet, Volume 1 of 2

Rev. 003

February 2025

Doc. No.: 833778, Rev.: 003



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.

Altering clock frequency, voltage, or memory interface speeds may void any product warranties and reduce stability, security, performance, and life of the processor and other components. Intel has not validated processor running memory above Plan-Of-Record (POR) speed. DRAM/DIMM devices should support desired speed, check with DRAM/DIMM vendors for details. System manufacturers are responsible for all validation and assume the risk of any stability, security, performance, or other functional issues resulting from such alterations

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

Copyright © 2024–2025, Intel Corporation. All rights reserved.

Contents

Revision History....12

1.0 Introduction....13

1.1 Overview....13

1.1.1 Terminology and Special Marks....15

1.2 SKU Definition....16

1.3 Flexible High Speed I/O....22

2.0 PCH Controller Device IDs....25

2.1 Revision IDs....25

2.2 Device IDs....25

3.0 Memory Mapping....28

3.1 Functional Description....28

3.1.1 PCI Devices and Functions....28

3.1.2 Fixed I/O Address Ranges....28

3.1.3 Variable I/O Decode Ranges....30

3.2 Memory Map....31

4.0 Audio Voice and Speech....34

4.1 Intel® High Definition Audio (Intel® HD Audio) Controller Capabilities....35

4.2 Audio DSP Capabilities....35

4.3 Intel® High Definition Audio Interface Capabilities....36

4.4 Direct Attached Digital Microphone (PDM) Interface....36

4.5 USB Audio Offload Support....37

4.6 I²S / PCM Interface....37

4.7 Intel® Display Audio Interface....37

4.8 MIPI® SoundWire* Interface....38

4.9 Signal Description....38

4.10 Integrated Pull-Ups and Pull-Downs....41

4.11 I/O Signal Planes and States....41

5.0 Ballout Definition....43

6.0 Connectivity Integrated (CNVi)....44

6.1 Functional Description....44

6.2 Signal Description....45

6.3 Integrated Pull-ups and Pull-downs....47

6.4 I/O Signal Planes and States....47

7.0 Controller Link....49

7.1 Integrated Pull-Ups and Pull-Downs....49

7.2 I/O Signal Planes and States....49

7.3 External CL_RST# Pin Driven/Open-drained Mode Support....49

7.4 Signal Description....50

8.0 Direct Enhanced Serial Peripheral Interface (Direct eSPI)....51

8.1 Functional Description....51

8.1.1 Configuration Flow....51

8.1.2 PCH Direct eSPI Registers....52

8.1.3 PCH Direct eSPI Reset Handling 52
8.1.4 PCH Direct eSPI Peripheral Channel Support 52
8.1.5 PCH Direct eSPI VW Channel Support 53
8.1.6 PCH Direct eSPI OOB Channel Support 53
8.1.7 PCH Direct eSPI Flash Channel Support 53
8.1.8 PCH eSPI Switch 54
8.1.9 Flash Descriptor Address Swapping 55
8.2 Protocols 56
8.3 Signal Description 57
8.4 Integrated Pull-Ups and Pull-Downs 57
8.5 I/O Signal Planes and States 58

9.0 Direct Media Interface (DMI) 59
9.1 Functional Description 59
9.1.1 Lane Reversal 59
9.1.2 Polarity Inversion 59
9.2 Signal Description 59
9.3 I/O Signal Planes and States 60
9.4 DMI Configuration 60
9.5 DMI CLKREQ, DMI REFCLK, DMI PERST# 61

10.0 Enhanced Serial Peripheral Interface (eSPI) 62
10.1 Functional Description 62
10.1.1 Operating Frequency 62
10.1.2 WAIT States from eSPI device 63
10.1.3 In-Band Link Reset 63
10.1.4 Flash Sharing Mode 63
10.1.5 PECI Over eSPI 63
10.1.6 Multiple OOB processes 63
10.1.7 Channels and Supported Transactions 63
10.2 Protocols 69
10.3 Signal Description 70
10.4 Integrated Pull-Ups and Pull-Downs 71
10.5 I/O Signal Planes and States 71

11.0 Gigabit Ethernet Controller 72
11.1 Signal Description 72
11.2 Integrated Pull-Ups and Pull-Downs 73
11.3 Functional Description 73
11.3.1 GbE PCI Bus Interface 75
11.3.2 Error Events and Error Reporting 76
11.3.3 Ethernet Interface 76
11.3.4 PCI Power Management 76

12.0 Host System Management Bus (SMBus) Controller 78
12.1 Functional Description 78
12.1.1 Host Controller 78
12.1.2 SMBus Target Interface 85
12.2 SMBus Power Gating 92
12.3 Signal Description 92
12.4 Integrated Pull-Ups and Pull-Downs 92
12.5 I/O Signal Planes and States 93

13.0 Integrated Sensor Hub (ISH)....94
13.1 Features....95
13.1.1 ISH I2C Controllers....95
13.1.2 ISH GSPI Controller....95
13.1.3 ISH UART Controller....95
13.1.4 ISH GPIOs....96
13.2 Functional Description....96
13.2.1 ISH Micro-Controller....96
13.2.2 SRAM....96
13.2.3 PCI Host Interface....96
13.2.4 ISH IPC....97
13.2.5 ISH Interrupt Handling via IOAPIC (Interrupt Controller)....97
13.3 Signal Description....97
13.4 Integrated Pull-Ups and Pull-Down....98
13.5 I/O Signal Planes and States....98
14.0 Intel® Serial I/O Improved Inter-Integrated Circuit (I3C) Controllers....100
14.1 Functional Description....101
14.1.1 Reset....101
14.1.2 Power Management....101
14.1.3 Interrupts....102
14.2 Signal Description....102
14.3 Integrated Pull-Ups and Pull-Downs....102
14.4 I/O Signal Planes and States....102
15.0 Intel® Serial I/O Inter-Integrated Circuit (I2C) Controllers....103
15.1 Functional Description....104
15.1.1 Protocols Overview....104
15.1.2 DMA Controller....105
15.1.3 Reset....106
15.1.4 Power Management....106
15.1.5 Interrupts....106
15.1.6 Error Handling....107
15.1.7 Programmable SDA Hold Time....107
15.2 Signal Description....107
15.3 Integrated Pull-Ups and Pull-Downs....107
15.4 I/O Signal Planes and States....108
16.0 Intel® Serial I/O Universal Asynchronous Receiver/Transmitter (UART) Controllers....109
16.1 Functional Description....110
16.1.1 UART Serial (RS-232) Protocols Overview....110
16.1.2 16550 8-bit Addressing - Debug Driver Compatibility....111
16.1.3 DMA Controller....111
16.1.4 Reset....112
16.1.5 Power Management....112
16.1.6 Interrupts....113
16.1.7 Error Handling....113
16.2 Signal Description....113
16.3 Integrated Pull-Ups and Pull-Downs....114
16.4 I/O Signal Planes and States....114

17.0 Intel® Serial IO Generic SPI (GSPI) Controllers..... 115
17.1 Functional Description.....115
17.1.1 Controller Overview..... 115
17.1.2 DMA Controller.....116
17.1.3 Reset.....117
17.1.4 Power Management..... 117
17.1.5 Interrupts.....118
17.1.6 Error Handling..... 118
17.2 Signal Description..... 118
17.3 Integrated Pull-Ups and Pull-Downs..... 119
17.4 I/O Signal Planes and States.....119

18.0 PCI Express* (PCIe*)..... 120
18.1 Functional Description.....120
18.1.1 PCI Express* Power Management..... 121
18.1.2 Port 80h Decode.....122
18.1.3 Advanced Error Reporting.....122
18.1.4 Single - Root I/O Virtualization (SR - IOV)..... 122
18.1.5 PCI Express* Receiver Lane Polarity Inversion.....123
18.1.6 Precision Time Measurement (PTM) 123
18.2 Signal Description..... 123
18.3 I/O Signal Planes and States.....123
18.4 PCI Express* Root Port Support Feature Details.....124

19.0 Real Time Clock (RTC).....127
19.1 Signal Description..... 127
19.2 I/O Signal Planes and States.....128

20.0 Serial ATA (SATA)..... 129
20.1 Functional Description.....129
20.1.1 Features Supported..... 129
20.1.2 SATA 6 Gb/s Support..... 130
20.1.3 Hot Plug Operation..... 130
20.1.4 Intel® Rapid Storage Technology (Intel® RST)..... 130
20.1.5 Power Management Operation.....132
20.1.6 SATA Device Presence..... 134
20.1.7 SATA LED..... 134
20.1.8 Advanced Host Controller Interface (AHCI) Operation..... 134
20.1.9 Enclosure Management (SGPIO Signals)..... 135
20.2 Signal Description..... 138
20.3 Integrated Pull-Ups and Pull-Downs..... 141
20.4 I/O Signal Planes and States.....141

21.0 Serial Peripheral Interface (SPI)..... 143
21.1 Functional Description.....143
21.1.1 SPI0 for Flash..... 143
21.1.2 SPI0 Support for TPM.....151
21.2 Signal Description..... 151
21.3 Integrated Pull-Ups and Pull-Downs..... 151
21.4 I/O Signal Planes and States.....152

22.0 System Clocks.....153
22.1 Integrated Clock Controller (ICC)..... 153
22.1.1 Signal Descriptions..... 153
22.2 I/O Signal Pin States.....154

23.0 System Management Interface and SMLink.....156
23.1 Functional Description.....156
23.1.1 Integrated USB-C* Usage.....156
23.2 Signal Description..... 157
23.3 Integrated Pull-Ups and Pull-Downs..... 158
23.4 I/O Signal Planes and States.....158

24.0 Touch Host Controller (THC)..... 159
24.1 Signal Description..... 159
24.2 Integrated Pull-Ups and Pull-Downs..... 159
24.3 I/O Signal Planes and States.....160
24.4 Functional Description.....160

25.0 Universal Serial Bus (USB).....162
25.1 Signal Description..... 162
25.2 Integrated Pull-Ups and Pull-Downs..... 166
25.3 I/O Signal Planes and States.....166
25.4 Supported USB 2.0 Ports.....167
25.5 Functional Description.....168
25.5.1 eXtensible Host Controller Interface (xHCI) Controller..... 168
25.5.2 USB Dual Role Support - eXtensible Device Controller Interface (xDCI) Controller.....168

26.0 General Purpose Input and Output.....169
26.1 Functional Description.....169
26.1.1 Configurable GPIO Voltage.....169
26.1.2 Interrupt / IRQ via GPIO Requirement.....170
26.1.3 Integrated Pull-ups and Pull-downs.....170
26.1.4 SCI / SMI and NMI..... 170
26.1.5 GPIO Blink (BK) and Serial Blink (SBK).....170
26.1.6 GPIO Ownership.....171
26.1.7 Native Function and TERM Bit Setting.....171
26.2 Signal Description..... 171

27.0 GPIO Serial Expander.....172
27.1 Signal Description..... 172
27.2 Integrated Pull-ups and Pull-downs.....172
27.3 Functional Description.....172

28.0 Private Configuration Space Port ID.....174

29.0 Security Firmware Engines.....175
29.1 Intel® Converged Security and Management Engine (Intel® CSME)..... 175
29.2 Intel® Silicon Security Engine..... 175

30.0 System Management.....176
30.1 Theory of Operation.....176
30.1.1 TCO Modes..... 176

31.0 Testability and Monitoring..... 179
31.1 Signal Description..... 179
31.2 I/O Signal Planes and States.....180
32.0 Miscellaneous Signals..... 181
32.1 Signal Description..... 181
32.2 Integrated Pull-Ups and Pull-Downs..... 182
32.3 Ground and Reserved Signals.....182
33.0 Processor Sideband Signals..... 184
33.1 Signal Description..... 184
33.2 I/O Signal Planes and States.....184
34.0 Power Delivery.....185
34.1 Power and Ground Signals.....185
35.0 Power Management..... 187
35.1 System Power States, Advanced Configuration and Power Interface (ACPI) 187
35.2 Legacy Power Management Support.....189
35.2.1 ALT Access Mode..... 190
35.2.2 Legacy Power Management Theory of Operation..... 190
35.3 Functional Description.....191
35.3.1 Features.....191
35.3.2 Power Saving Features 192
35.3.3 SMI#/SCI Generation.....192
35.3.4 Sleep States..... 194
35.3.5 Event Input Signals and Their Usage..... 199
35.3.6 System Power Supplies, Planes, and Signals.....202
35.3.7 Reset Behavior.....207
35.3.8 Legacy Power Management Theory of Operation..... 209
35.4 Power and Internal Signals..... 209
35.4.1 Signal Description.....210
35.4.2 Integrated Pull-Ups and Pull-Downs..... 212
35.4.3 I/O Signal Planes and States.....212
36.0 Thermal Sensor.....214
36.1 Modes of Operation.....214
36.2 Temperature Trip Point..... 214
36.3 Thermal Sensor Accuracy ($T_{accuracy}$)..... 215
36.4 Thermal Reporting to EC..... 215
36.5 Thermal Trip Signal (PCHHOT#)..... 215
37.0 Electrical Characteristics 216

Figures

1 Flexible HSIO Lane Details 23
2 Processor/PCH Descriptor Layout/Swapping 56
3 Basic eSPI Protocol..... 56
4 eSPI Device Request to PCH for PCH Temperature..... 66
5 PCH Response to eSPI device with PCH Temperature 66
6 eSPI Device Request to PCH for PCH RTC Time..... 67
7 PCH Response to eSPI device with RTC Time 68
8 Basic eSPI Protocol..... 69
9 Data Transfer on I^2C Bus..... 104
10 UART Serial Protocol 110
11 UART Receiver Serial Data Sample Points..... 111
12 Supported PCI Express* Link Configurations 125
13 Port Enable/Device Present Bits Flow..... 134
14 Serial Data Transmitted over SGPIO Interface..... 137
15 Flash Descriptor Regions..... 146
16 Flash Descriptor Redundancy..... 149
17 ICC Diagram..... 153
18 THC Block Diagram..... 161
19 Supported USB 2.0 Ports on Intel® 800 Series Chipset..... 168
20 GSX Topology - Example..... 173
21 TCO Compatible Mode SMBus Configuration..... 177
22 Advanced TCO Mode..... 178
23 Conceptual Diagram of SLP_LAN#..... 205

Tables

1 References 14
2 PCH I/O Capabilities 14
3 Terminology 15
4 Special Marks 15
5 Desktop PCH SKUs 16
6 Mobile PCH SKUs 17
7 Workstation PCH SKUs 17
8 Desktop PCH HSIO Details 18
9 Mobile PCH HSIO Details 20
10 Workstation PCH HSIO Details 21
11 Acronyms 23
12 Device and Revision ID 25
13 PCH ACPI Device ID for GPIO Controller 27
14 Fixed I/O Ranges Decoded by PCH 28
15 Variable I/O Decode Ranges 30
16 PCH Memory Decode Ranges (Processor Perspective) 31
17 Acronyms 34
18 References 35
19 Integrated Pull-Ups and Pull-Downs 41
20 I/O Signal Planes and States 41
21 Acronyms 44
22 References 44
23 Acronyms 49
24 Acronyms 51
25 References 51
26 PCH DMI Link Mapping 60
27 Acronyms 62
28 References 62
29 eSPI Channels and Supported Transactions 64
30 eSPI Virtual Wires (VW) 64
31 Acronyms 72
32 References 72
33 GbE LAN Signals 72
34 LAN Mode Support 76
35 Acronyms 78
36 References 78
37 I^2C* Block Read 82
38 Enable for SMBALERT# 84
39 Enables for SMBus Target Write and SMBus Host Events 84
40 Enables for the Host Notify Command 84
41 Target Write Registers 86
42 Command Types 86
43 Target Read Cycle Format 87
44 Data Values for Target Read Registers 87
45 Host Notify Format 89
46 Target Read Cycle Format 90
47 Data Values for Target Read Registers 90
48 Enables for SMBus Target Write and SMBus Host Events 92
49 Acronyms 94
50 References 95
51 IPC Initiator -> Target flows 97
52 Acronyms 101
53 Acronyms 104
54 References 104

55 Acronyms..... 110
56 Acronyms..... 115
57 Acronym..... 120
58 Reference..... 120
59 Features Supported..... 120
60 Power Plane and States for PCI Express* Signals 123
61 PCI Express* Root Port Feature Details 124
62 Acronyms..... 127
63 Acronyms..... 129
64 References..... 129
65 Acronyms..... 143
66 SPI0 Flash Regions..... 144
67 Region Size Versus Erase Granularity of Flash Components 144
68 Region Access Control Table..... 147
69 Flash Descriptor Processor Complex Soft Strap..... 148
70 Signal Descriptions..... 153
71 I/O Signal Pin States..... 154
72 Acronyms..... 156
73 Acronyms..... 159
74 Acronyms..... 162
75 References..... 162
76 Acronyms..... 169
77 Native Function Signals Supporting Dynamic Termination Override..... 171
78 Private Configuration Space Register Target Port IDs 174
79 Acronyms..... 176
80 Event Transitions that Cause Messages..... 177
81 Acronyms..... 179
82 References..... 179
83 Testability Signals..... 179
84 Testability Signals..... 180
85 Power Planes and States for Testability Signals..... 180
86 Signal Descriptions..... 181
87 Integrated Pull-Ups and Pull-Downs..... 182
88 GND and RSVD Signals..... 183
89 Acronyms..... 184
90 Power Rail Descriptions..... 185
91 Acronyms..... 187
92 References..... 187
93 General System Power States 187
94 State Transition Rules for the PCH..... 188
95 System Power Plane..... 189
96 Register Write Accesses in ALT Access Mode..... 190
97 Write Only Registers with Read Paths in ALT Access Mode..... 190
98 Causes of SMI and SCI 192
99 Sleep Types 195
100 Causes of Wake Events..... 195
101 Transitions Due to Power Failure 197
102 Supported Deep Sx Policy Configurations 198
103 Deep Sx Wake Events 199
104 Transitions Due to Power Button..... 200
105 PRIMPWRDNACK/PRIMWARN#/GPP_A02 Pin Behavior..... 206
106 PRIMPWRDNACK During Reset..... 206
107 Causes of Host and Global Resets..... 207

Revision History

Document Number	Revision Number	Description	Revision Date
833778	001	Initial Release for Z890 chipset	October 2024
833778	002	Initial Release for H810, B860, Q870, W880, WM880, HM870 chipsets	January 2025
833778	003	Introduction on page 13 • **Updated** WM880 and HM870 SKU information for Integrated Thunderbolt Lanes in Table 6 on page 17	February 2025

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® 800 Series Chipset Family Platform Controller Hub (PCH).

- Throughout this document, the Platform Controller Hub (PCH) is used as a general term and refers to all PCH SKUs, unless specifically noted otherwise.
- Throughout this document, PCH refers to desktop, workstation, and mobile segment PCH SKUs, unless specifically noted otherwise.
- Throughout this document, the terms "Desktop" and "Desktop Only" refers to information that is applicable only to Desktop PCH, unless specifically noted otherwise.
- Throughout this document, the terms "Workstation" and "Workstation Only" refers to information that is applicable only to Workstation PCH, unless specifically noted otherwise.
- Throughout this document, the terms "Mobile" and "Mobile Only" refers to information that is applicable only to Mobile PCH, unless specifically noted otherwise.

This document 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 document abbreviates buses as Bn, devices as Dn and functions as Fn. For example, Device 31 Function 0 is abbreviated as D31:F0, Bus 1 Device 8 Function 0 is abbreviated as B1:D8:F0. The bus number for the PCH is assigned by BIOS and is represented by the variable M in this document.

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 5.0
- Integrated Serial ATA Host controller 3.2, supports data transfer rates of up to 6 Gb/s on all ports
- USB 3.2 Gen 2x2 (20 Gb/s) eXtensible Host Controller (xHCI)
- USB 3.2 Gen 1x1 (5 Gb/s) Dual Role (eXtensible Device Controller - xDCI) Capability
- Direct Media Interface (DMI)
- Serial Peripheral Interface (SPI)
- Enhanced Serial Peripheral Interface (eSPI)
- General Purpose Input Output (GPIO)

- System Management Bus (SMBus) Specification, Version 2.0
- Integrated Clock Controller (ICC)/Real Time Clock Controller (RTCC)
- Intel® High Definition Audio (Intel® HD Audio) and Intel® Smart Sound Technology (Intel® SST), supporting I^2S, MIPI* SoundWire*, and DMIC.
- Intel® Serial I/O UART Host controllers
- Intel® Serial I/O I^2C Host controllers
- Intel® Serial I/O Improved Inter-Integrated Circuit (I^3C) 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 4.8
- Supports Integrated connectivity (CNVi)
- Direct eSPI

Table 1. References

Specification	Document Number
Intel® 800 Series Chipset Family Platform Controller Hub Datasheet, Volume 2 of 2	834576

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	Intel® 800 Series Chipset
DMI	Gen4 x8
PCIe* Gen4 Lanes	24
PCIe Reference Clock Outputs	14
SATA	8, Gen3, 4-port mux granularity Support 2 M.2 with SATA
Integrated GbE Controller	1 Data Link to Intel® Ethernet connection I219
Maximum USB 3.2 Gen 2x1 (10 Gb/s)	10
Maximum USB 3.2 Gen 2x2 (20 Gb/s)	5
Maximum USB 2.0 Ports	14

continued...

Interface	Intel® 800 Series Chipset
Audio	Intel® HD Audio, 3 I^2S, 2 DMIC, 4x1 SoundWire*
eSPI	4 CS#, Quad Mode
Integrated Sensor Hub (ISH)	ISH 5.6
ISH IO	1 SPI, 3 I^2C, 1 I^3C, 2 UART, 12 GPIO
GPIO	1.8 V or 3.3 V GPIO per family
GPIO Expander Support	YES

1.1.1 Terminology and Special Marks

Table 3. Terminology

Term	Description
DMA	Direct Memory Access
HDMI*	High Definition Multimedia Interface
Intel® TH	Intel® Trace Hub
IOV	I/O Virtualization
LTR	The Latency Tolerance Reporting (LTR) mechanism enables endpoints to report their service latency requirements for Memory Reads and Writes to the Root Complex, so that power management policies for central platform resources (such as main memory, RC internal interconnects, and snoop resources) can be implemented to consider endpoint service requirements.
PECI	Platform Environment Control Interface
Processor	The 64-bit multi-core component (package)
S0ix-states	Processor and PCH residency idle standby power states.
SCI	System Control Interrupt. SCI is used in the ACPI protocol.
SHA	Secure Hash Algorithm
SSC	Spread Spectrum Clock
STR	Suspend to RAM
V_{CC}	PCH Core Power Supply
V_{SS}	PCH Ground

Table 4. Special Marks

Mark	Definition
#	A suffix of # indicates an active low signal. For example, RSMRST#.
h	Hexadecimal numbers are identified with an h in the number. All numbers are decimal (base 10) unless otherwise specified. Non-obvious binary numbers have the 'b' enclosed at the end of the number. For example, 0101b.

1.2 SKU Definition

Table 5. Desktop PCH SKUs

Features	H810	B860	Q870	Z890
PCH IO Capabilities				
DMI	Gen 4 x4	Gen 4 x4	Gen 4 x8	Gen 4 x8
Maximum SATA 6 Gb/s Ports	4	4	8	8
Maximum PCIe* Gen4 Lanes	8	14	20	24
Total USB 2.0 Ports	10	12	14	14
Maximum USB 3.2 Gen 1x1 (5 Gb/s)	4	6	10	10
Maximum USB 3.2 Gen 2x1 (10 Gb/s)	2	4	8	10
Maximum USB 3.2 Gen 2x2 (20 Gb/s)	0	2	4	5
Intel® CSME Firmware	Intel® CSME 19 Consumer	Intel® CSME 19 Consumer	Intel® CSME 19 Corporate	Intel® CSME 19 Consumer
eSPI Chip Select	2	2	4	4
PCIe* RAID 0, 1, 5, 10 Support	NO	NO	YES	YES
SATA RAID 0, 1, 5, 10 Support	YES	YES	YES	YES
Processor IO Capabilities with PCH Pairing [1]				
Processor Thunderbolt™ 4 / USB4* Ports	1	1	Up to 2	Up to 2
IA & BCLK Overclocking	NO	NO	NO	YES
Memory Overclocking	NO	YES	NO	YES
Processor PCI Express 5.0 Lanes Configuration Support	1 x 16	1 x 16 + 1 x 4	1 x 16 + 1 x 4 or 2 x 8 + 1 x 4 or 1 x 8 + 3 x 4	1 x 16 + 1 x 4 or 2 x 8 + 1 x 4 or 1 x 8 + 3 x 4
Processor PCI Express 4.0 Lanes Configuration Support	NO	NO	1 x 4	1 x 4
System Memory Channels/DPC	2/1	2/2	2/2	2/2
ECC Memory	NO	NO	NO	NO
Number of Simultaneous Independent Displays Supported	3	4	4	4

Note: 1. These are not features of the PCH, but rather pertain to the processor when paired with a specific PCH SKU.

Table 6. Mobile PCH SKUs

Features	WM880	HM870
PCH IO Capabilities		
DMI	DMI Gen 4 x8	DMI Gen 4 x8
Maximum SATA 6 Gb/s Ports	8	8
Maximum PCIe* Gen4 Lanes	24	24
Total USB 2.0 Ports	14	14
Maximum USB 3.2 Gen 1x1 (5 Gb/s)	10	10
Maximum USB 3.2 Gen 2x1 (10 Gb/s)	10	10
Maximum USB 3.2 Gen 2x2 (20 Gb/s)	5	5
Intel® CSME Firmware	Intel® CSME 19 Consumer and Corporate	Intel® CSME 19 Consumer
eSPI Chip Select	4	4
PCIe* RAID 0, 1, 5, 10 Support	YES	YES
SATA RAID 0, 1, 5, 10 Support	YES	YES
Processor IO Capabilities with PCH Pairing [1]		
Integrated Thunderbolt Lanes	Up to 2	Up to 2
IA & BCLK Overclocking	NO	YES
Memory Overclocking	NO	YES
Processor PCI Express 5.0 Lanes Configuration Support	1 x 16 + 1 x 4 or 2 x 8 + 1 x 4 or 1 x 8 + 3 x 4	1 x 16 + 1 x 4 or 2 x 8 + 1 x 4 or 1 x 8 + 3 x 4
Processor PCI Express 4.0 Lanes Configuration Support	1 x 4	1 x 4
System Memory Channels/DPC	2	2
ECC Memory	YES	NO
Number of Simultaneous Independent Displays Supported	4	4

Note: 1. These are not features of the PCH, but rather pertain to the processor when paired with a specific PCH SKU.

Table 7. Workstation PCH SKUs

Features	W880 (entry-level workstation)
PCH IO Capablities	
DMI	DMI Gen 4 x8
Maximum SATA 6 Gb/s Ports	8
Maximum PCIe* Gen4 Lanes	24
Total USB 2.0 Ports	14
Maximum USB 3.2 Gen 1x1 (5 Gb/s)	10

continued...

Features	W880 (entry-level workstation)
Maximum USB 3.2 Gen 2x1 (10 Gb/s)	10
Maximum USB 3.2 Gen 2x2 (20 Gb/s)	5
Intel® CSME Firmware	Intel® CSME 19 Corporate
eSPI Chip Select	4
PCIe* RAID 0, 1, 5, 10 Support	YES
SATA RAID 0, 1, 5, 10 Support	YES
Processor IO Capabilities with PCH Pairing [1]	
Processor Thunderbolt™ 4 / USB4* Ports	Up to 2
IA & BCLK Overclocking	NO
Memory Overclocking	YES
Processor PCI Express 5.0 Lanes Configuration Support	1 x 16 + 1 x 4 or 2 x 8 + 1 x 4 or 1 x 8 + 3 x 4
Processor PCI Express 4.0 Lanes Configuration Support	1 x 4
System Memory Channels/DPC	2/2
ECC Memory	YES
Number of Simultaneous Independent Displays Supported	4
Note: 1. These are not features of the PCH, but rather pertain to the processor when paired with as specific PCH SKU.	

Table 8. Desktop PCH HSIO Details

Flex I/O Lane	H810	B860 [1]		Q870		Z890	
0	USB 3.2 Gen 2x1 #1	USB 3.2 Gen 2x1 #1	USB 3.2 Gen 2x2 #1	USB 3.2 Gen 2x1 #1	USB 3.2 Gen 2x2 #1	USB 3.2 Gen 2x1 #1	USB 3.2 Gen 2x2 #1
1	USB 3.2 Gen 2x1 #2	USB 3.2 Gen 2x1 #2		USB 3.2 Gen 2x1 #2		USB 3.2 Gen 2x1 #2	
2	USB 3.2 Gen 1x1 #3	USB 3.2 Gen 2x1 #3	USB 3.2 Gen 2x2 #2	USB 3.2 Gen 2x1 #3	USB 3.2 Gen 2x2 #2	USB 3.2 Gen 2x1 #3	USB 3.2 Gen 2x2 #2
3	USB 3.2 Gen 1x1 #4	USB 3.2 Gen 2x1 #4		USB 3.2 Gen 2x1 #4		USB 3.2 Gen 2x1 #4	
4	Not Available	Not Available		USB 3.2 Gen 1x1 #5		USB 3.2 Gen 2x1 #5	USB 3.2 Gen 2x2 #3
5	Not Available	Not Available		USB 3.2 Gen 1x1 #6		USB 3.2 Gen 2x1 #6	
6	Not Available	Not Available		USB 3.2 Gen 2x1 #7	USB 3.2 Gen 2x2 #4	USB 3.2 Gen 2x1 #7	USB 3.2 Gen 2x2 #4

continued...

Flex I/O Lane	H810	B860 [1]	Q870		Z890	
7	Not Available	Not Available	USB 3.2 Gen 2x1 #8		USB 3.2 Gen 2x1 #8	
8	Not Available	USB 3.2 Gen 1x1 #9	USB 3.2 Gen 2x1 #9	USB 3.2 Gen 2x2 #5	USB 3.2 Gen 2x1 #9	USB 3.2 Gen 2x2 #5
9	Not Available	USB 3.2 Gen 1x1 #10	USB 3.2 Gen 2x1 #10		USB 3.2 Gen 2x1 #10	
10	PCIe* #1	PCIe #1	PCIe #1		PCIe #1	
11	PCIe #2	PCIe #2	PCIe #2		PCIe #2	
12	PCIe #3;GbE	PCIe #3;GbE	PCIe #3;GbE		PCIe #3;GbE	
13	PCIe #4	PCIe #4	PCIe #4		PCIe #4	
14	PCIe #5	PCIe #5	PCIe #5		PCIe #5	
15	PCIe #6	PCIe #6	PCIe #6		PCIe #6	
16	PCIe #7;GbE	PCIe #7;GbE	PCIe #7;GbE		PCIe #7;GbE	
17	PCIe #8	PCIe #8	PCIe #8		PCIe #8	
18	Not Available	Not Available	Not Available		PCIe #9	
19	Not Available	Not Available	Not Available		PCIe #10	
20	Not Available	Not Available	Not Available		PCIe #11	
21	Not Available	Not Available	Not Available		PCIe #12	
22	Not Available	Not Available	PCIe #13;SATA #0		PCIe #13;SATA #0	
23	Not Available	Not Available	PCIe #14;SATA #1		PCIe #14;SATA #1	
24	GbE	PCIe #15;GbE	PCIe #15;SATA #2;GbE		PCIe #15;SATA #2;GbE	
25	Not Available	PCIe #16	PCIe #16;SATA #3		PCIe #16;SATA #3	
26	SATA #4	SATA #4	PCIe #17;SATA #4		PCIe #17;SATA #4	
27	SATA #5	SATA #5	PCIe #18;SATA #5		PCIe #18;SATA #5	
28	SATA #6	SATA #6	PCIe #19;SATA #6		PCIe #19;SATA #6	
29	SATA #7	SATA #7	PCIe #20;SATA #7		PCIe #20;SATA #7	
30	Not Available	PCIe #21	PCIe #21		PCIe #21	
31	Not Available	PCIe #22	PCIe #22		PCIe #22	
32	Not Available	PCIe #23	PCIe #23		PCIe #23	
33	Not Available	PCIe #24	PCIe #24		PCIe #24	
34	Not Available	Not Available	DMI #7		DMI #7	
35	Not Available	Not Available	DMI #6		DMI #6	
36	Not Available	Not Available	DMI #5		DMI #5	
37	Not Available	Not Available	DMI #4		DMI #4	
38	DMI #3	DMI #3	DMI #3		DMI #3	

continued...

Flex I/O Lane	H810	B860 [1]	Q870	Z890
39	DMI #2	DMI #2	DMI #2	DMI #2
40	DMI #1	DMI #1	DMI #1	DMI #1
41	DMI #0	DMI #0	DMI #0	DMI #0

NOTE

1. For the PCH-S B860 SKU, the PCIe Controller #4 associated PCIe Lanes 13 and 14 are not available. If used this Controller must only route PCIe Lanes 15 and/or 16 and the Controller configuration mode must be set accordingly to any configuration other than 1x4 or 1x4 Lane Reversal.

Table 9. Mobile PCH HSIO Details

Flex I/O Lane	WM880		HM870	
0	USB 3.2 Gen 2x1 #1	USB 3.2 Gen 2x2 #1	USB 3.2 Gen 2x1 #1	USB 3.2 Gen 2x2 #1
1	USB 3.2 Gen 2x1 #2		USB 3.2 Gen 2x1 #2	
2	USB 3.2 Gen 2x1 #3	USB 3.2 Gen 2x2 #2	USB 3.2 Gen 2x1 #3	USB 3.2 Gen 2x2 #2
3	USB 3.2 Gen 2x1 #4		USB 3.2 Gen 2x1 #4	
4	USB 3.2 Gen 2x1 #5	USB 3.2 Gen 2x2 #3	USB 3.2 Gen 2x1 #5	USB 3.2 Gen 2x2 #3
5	USB 3.2 Gen 2x1 #6		USB 3.2 Gen 2x1 #6	
6	USB 3.2 Gen 2x1 #7	USB 3.2 Gen 2x2 #4	USB 3.2 Gen 2x1 #7	USB 3.2 Gen 2x2 #4
7	USB 3.2 Gen 2x1 #8		USB 3.2 Gen 2x1 #8	
8	USB 3.2 Gen 2x1 #9	USB 3.2 Gen 2x2 #5	USB 3.2 Gen 2x1 #9	USB 3.2 Gen 2x2 #5
9	USB 3.2 Gen 2x1 #10		USB 3.2 Gen 2x1 #10	
10	PCIe* #1		PCIe #1	
11	PCIe #2		PCIe #2	
12	PCIe #3;GbE		PCIe #3;GbE	
13	PCIe #4		PCIe #4	
14	PCIe #5		PCIe #5	
15	PCIe #6		PCIe #6	
16	PCIe #7;GbE		PCIe #7;GbE	
17	PCIe #8		PCIe #8	
18	PCIe #9		PCIe #9	
19	PCIe #10		PCIe #10	
20	PCIe #11		PCIe #11	
21	PCIe #12		PCIe #12	
22	PCIe #13;SATA #0		PCIe #13;SATA #0	
23	PCIe #14;SATA #1		PCIe #14;SATA #1	

continued...

Flex I/O Lane	WM880	HM870
24	PCIe #15;SATA #2;GbE	PCIe #15;SATA #2;GbE
25	PCIe #16;SATA #3	PCIe #16;SATA #3
26	PCIe #17;SATA #4	PCIe #17;SATA #4
27	PCIe #18;SATA #5	PCIe #18;SATA #5
28	PCIe #19;SATA #6	PCIe #19;SATA #6
29	PCIe #20;SATA #7	PCIe #20;SATA #7
30	PCIe #21	PCIe #21
31	PCIe #22	PCIe #22
32	PCIe #23	PCIe #23
33	PCIe #24	PCIe #24
34	DMI #7	DMI #7
35	DMI #6	DMI #6
36	DMI #5	DMI #5
37	DMI #4	DMI #4
38	DMI #3	DMI #3
39	DMI #2	DMI #2
40	DMI #1	DMI #1
41	DMI #0	DMI #0

Table 10. Workstation PCH HSIO Details

Flex I/O Lane	W880 (entry-level workstation)	
0	USB 3.2 Gen 2x1 #1	USB 3.2 Gen 2x2 #1
1	USB 3.2 Gen 2x1 #2	
2	USB 3.2 Gen 2x1 #3	USB 3.2 Gen 2x2 #2
3	USB 3.2 Gen 2x1 #4	
4	USB 3.2 Gen 2x1 #5	USB 3.2 Gen 2x2 #3
5	USB 3.2 Gen 2x1 #6	
6	USB 3.2 Gen 2x1 #7	USB 3.2 Gen 2x2 #4
7	USB 3.2 Gen 2x1 #8	
8	USB 3.2 Gen 2x1 #9	USB 3.2 Gen 2x2 #5
9	USB 3.2 Gen 2x1 #10	
10	PCIe* #1	
11	PCIe #2	
12	PCIe #3;GbE	
13	PCIe #4	
14	PCIe #5	

continued...

Flex I/O Lane	W880 (entry-level workstation)
15	PCIe #6
16	PCIe #7;GbE
17	PCIe #8
18	PCIe #9
19	PCIe #10
20	PCIe #11
21	PCIe #12
22	PCIe #13;SATA #0
23	PCIe #14;SATA #1
24	PCIe #15;SATA #2;GbE
25	PCIe #16;SATA #3
26	PCIe #17;SATA #4
27	PCIe #18;SATA #5
28	PCIe #19;SATA #6
29	PCIe #20;SATA #7
30	PCIe #21
31	PCIe #22
32	PCIe #23
33	PCIe #24
34	DMI #7
35	DMI #6
36	DMI #5
37	DMI #4
38	DMI #3
39	DMI #2
40	DMI #1
41	DMI #0

1.3 Flexible High Speed I/O

Flexible Input/Output (I/O) is a technology that allows the 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 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 Introduction on page 13 for specific SKU implementation details.

Table 11. Acronyms

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

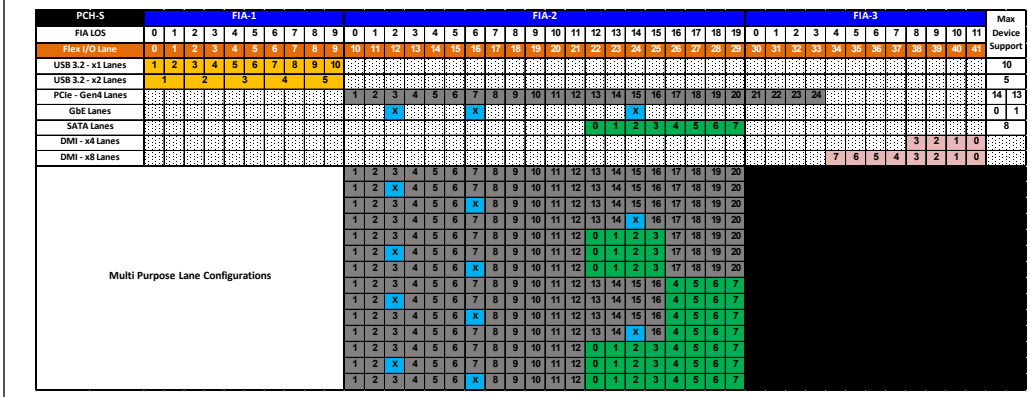
Flexible I/O Lane Selection

HSIO lane configuration and type are statically selected by soft straps.

NOTE

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

Figure 1. Flexible HSIO Lane Details



The PCH Flexible I/O Lane support will vary depending on the PCH SKU. Refer to Introduction for specific SKU implementation details

FIA = Flex-IO Adapter

FIA LOS = Flex-IO Adapter Lane Ownership Number

Multi Purpose Lane Configuration details cover the Flexible HSIO Configuration Options

The 42 Flexible HSIO Lanes [41:0] support the following:

1. Up to twenty four PCI Express* (PCIe) Lanes
   - A maximum of fourteen PCIe Root Ports can be enabled when GbE Port is disabled
   - A maximum of thirteen PCIe Root Ports can be enabled when GbE Port is enabled
   - PCIe Lanes 1-4 (PCIe Controller #1 Gen4), 5-8 (PCIe Controller #2 Gen4), 9-12 (PCIe Controller #3 Gen4), 13-16 (PCIe Controller #4 Gen4), 17-20 (PCIe Controller #5 Gen4), and 21-24 (PCIe Controller #6 Gen4) must be individually configured
2. Up to ten USB 3.2 Gen 1x1/2x1 Lanes
   - A maximum of ten USB 3.2 Gen 1x1/2x1 Ports can be enabled
   - USB 3.2 Gen 1x1 = First Generation with One 5 GT/s Data Lane
   - USB 3.2 Gen 2x1 = Second Generation with One 10 GT/s Data Lane
3. Up to five USB 3.2 Gen 2x2 Lanes
   - A maximum of five USB 3.2 Gen 2x2 Ports can be enabled
   - USB 3.2 Gen 2x2 = Second Generation with Two 10 GT/s Data Lane
4. Up to eight SATA Lanes
   - A maximum of eight SATA Ports can be enabled
5. Up to three GbE Lanes
   - A maximum of one GbE Port can be enabled
6. Up to eight DMI Lanes for x4 and x8 support

2.0 PCH Controller Device IDs

2.1 Revision IDs

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

2.2 Device IDs

This section specifies the device IDs of the PCH.

Table 12. Device and Revision ID

Dev ID	Bus Number	Device Function - Device Description	B0 SRID	Note
7F00h - 7F1Fh	M	D31 : F0 - eSPI Controller	10h	**PCH Device IDs** **Desktop** H810 : 7F07h B860 : 7F06h Q870 : 7F03h Z890 : 7F04h **Mobile** WM880 : 7F0Dh HM870 : 7F0Ch **Workstation entry-level** W880 : 7F08h
7F20h	M	D31 : F1 - P2SB	10h	
7F21h	M	D31 : F2 - PMC	10h	
7F23h	M	D31 : F4 - SMBus	10h	
7F24h	M	D31 : F5 - SPI (flash) Controller	10h	
550Ch	M	D31 : F6 - GbE Controller: Corporate/Intel® vPro™ (Default)	10h	
550Dh	M	D31 : F6 - GbE Controller: Consumer	10h	
7F26h	M	D31 : F7 - Intel® Trace Hub (Intel® TH)	10h	
7F27h	M	D20 : F2 - Shared SRAM	10h	
7F28h	M	D30 : F0 - UART #0	10h	
7F29h	M	D30 : F1 - UART #1	10h	
7F2Ah	M	D30 : F2 - GSPI #0	10h	
7F2Bh	M	D30 : F3 - GSPI #1	10h	
7F30h	M	D29 : F0 - PCI* Express Root Port #9	10h	
7F31h	M	D29 : F1 - PCI Express Root Port #10	10h	

continued...

Dev ID	Bus Number	Device Function - Device Description	B0 SRID	Note
7F32h	M	D29 : F2 - PCI Express Root Port #11	10h	
7F33h	M	D29 : F3 - PCI Express Root Port #12	10h	
7F34h	M	D29 : F4 - PCI Express Root Port #13	10h	
7F35h	M	D29 : F5 - PCI Express Root Port #14	10h	
7F36h	M	D29 : F6 - PCI Express Root Port #15	10h	
7F37h	M	D29 : F7 - PCI Express Root Port #16	10h	
7F38h	M	D28 : F0 - PCI Express Root Port #1	10h	
7F39h	M	D28 : F1 - PCI Express Root Port #2	10h	
7F3Ah	M	D28 : F2 - PCI Express Root Port #3	10h	
7F3Bh	M	D28 : F3 - PCI Express Root Port #4	10h	
7F3Ch	M	D28 : F4 - PCI Express Root Port #5	10h	
7F3Dh	M	D28 : F5 - PCI Express Root Port #6	10h	
7F3Eh	M	D28 : F6 - PCI Express Root Port #7	10h	
7F3Fh	M	D28 : F7 - PCI Express Root Port #8	10h	
7F40h	M	D27 : F0 - PCI Express Root Port #17	10h	
7F41h	M	D27 : F1 - PCI Express Root Port #18	10h	
7F42h	M	D27 : F2 - PCI Express Root Port #19	10h	
7F43h	M	D27 : F3 - PCI Express Root Port #20	10h	
7F44h	M	D27 : F4 - PCI Express Root Port #21	10h	
7F45h	M	D27 : F5 - PCI Express Root Port #22	10h	
7F46h	M	D27 : F6 - PCI Express Root Port #23	10h	
7F47h	M	D27 : F7 - PCI Express Root Port #24	10h	
7F4Ch	M	D21 : F0 - I^2C Controller #0	10h	
7F4Dh	M	D21 : F1 - I^2C Controller #1	10h	
7F4Eh	M	D21 : F2 - I^2C Controller #2	10h	
7F4Fh	M	D21 : F3 - I^2C Controller #3	10h	
7F50h	M	D31 : F3 - ACE (Audio Context Engine)	10h	
7F58h	M	D16 : F0 - THC #0 (Touch Host Controller) ID1	10h	
7F59h	M	D16 : F0 - THC #0 (Touch Host Controller) ID2	10h	
7F5Ah	M	D16 : F1 - THC #1 (Touch Host Controller) ID1	10h	
7F5Bh	M	D16 : F1 - THC #1 (Touch Host Controller) ID2	10h	
7F5Ch	M	D25 : F2 - UART #2	10h	
7F5Dh	M	D19 : F1 - UART #3	10h	

continued...

Dev ID	Bus Number	Device Function - Device Description	B0 SRID	Note
7F5Eh	M	D18 : F6 - GSPI #2	10h	
7F5Fh	M	D19 : F0 - GSPI #3	10h	
7F62h	M	D23 : F0 - SATA Controller (AHCI) - Desktop	10h	
7F66h	M	D23 : F0 - SATA Controller (RAID 0/1/5/10) - premium - Desktop [AIE=1]	10h	This SATA DID is applicable to HEDT/WS SKU only
7F68h	M	D22 : F0 - HECI #1	10h	
7F69h	M	D22 : F1 - HECI #2	10h	
7F6Ah	M	D22 : F2 - IDE Redirection (IDER-R)	10h	
7F6Bh	M	D22 : F3 - Keyboard and Text (KT) Redirection	10h	
7F6Ch	M	D22 : F4 - HECI #3	10h	
7F6Dh	M	D22 : F5 - HECI #4	10h	
7F6Eh	M	D20 : F0 - USB 3.1 xHCI HC	10h	
7F6Fh	M	D20 : F1 - USB Device Controller (OTG) (xDCI)	10h	
7F70h	M	D20 : F3 - CNVi: Wi-Fi*	10h	
7F78h	M	D18 : F0 - Integrated Sensor Hub	10h	
7F79h	M	D21 : F4 - I^3C Controller	10h	
7F7Ah	M	D25 : F0 - I^2C Controller #4	10h	
7F7Bh	M	D25 : F1 - I^2C Controller #5	10h	
7F7Ch	M	D24 : F0 - Intel® Silicon Security Engine HECI #1	10h	
7F7Dh	M	D24 : F1 - Intel® Silicon Security Engine HECI #2	10h	
7F7Eh	M	D24 : F2 - Intel® Silicon Security Engine HECI #3	10h	

Table 13. PCH ACPI Device ID for GPIO Controller

ACPI ID
INTC1084

3.0 Memory Mapping

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

3.1 Functional Description

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. 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.

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.

I/O cycles that go to target ranges that are marked as Reserved will be handled as follows:

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 (unless assigned to one of the variable ranges) and will be internally terminated.

Table 14. Fixed I/O Ranges Decoded by PCH

I/O Address	Read Target	Write Target	Internal Unit (Unless[E]: External)[2]	Separate Enable/Disable
2E-2F	Super I/O	Super I/O	[E] Forwarded to eSPI	Yes. ESPI_IOD_IOE.SE
4E-4F	Microcontroller	Microcontroller	[E] Forwarded to eSPI	Yes. ESPI_IOD_IOE.ME2
60h	Keyboard Controller	Keyboard Controller	[E] Forwarded to eSPI	Yes, with 64h. ESPI_IOD_IOE.KE
62h	Microcontroller	Microcontroller	[E] Forwarded to eSPI	Yes, with 66h.

continued...

I/O Address	Read Target	Write Target	Internal Unit (Unless[E]: External)[2]	Separate Enable/Disable
				ESPI_IOD_IOE.ME1
64h	Keyboard Controller	Keyboard Controller	[E] Forwarded to eSPI	Yes, with 60h. ESPI_IOD_IOE.KE
66h	Microcontroller	Microcontroller	[E] Forwarded to eSPI	Yes, with 62h. ESPI_IOD_IOE.ME1
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=0, else 72h
73h	RTC Controller	RTC Controller	RTC	None. Alias to 71h if RC.UE='0', else 73h
74h	RTC Controller	RTC Controller	RTC	None
75h	RTC Controller	RTC Controller	RTC	None
76h-77h	RTC Controller	RTC Controller	RTC	None. Alias to 70h-71h if RC.UE=0, else 76h-77h
80h[3]	eSPI	eSPI	Read: [E] eSPI Write: [E] eSPI	None. eSPI if GCS.RPR='0'
84h - 86h	eSPI	eSPI	Read: [E] eSPI Write: [E] eSPI	None. eSPI if GCS.RPR='0'
88h	eSPI	eSPI	Read: [E] eSPI Write: [E] eSPI	None. eSPI if GCS.RPR='0'
8Ch - 8Eh	eSPI	eSPI	Read: [E] eSPI Write: [E] eSPI	None. eSPI if GCS.RPR='0'
90h	eSPI	eSPI	Read: [E] eSPI Write: [E] eSPI	None. Alias to 80h
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
200-207h	Gameport Low	Gameport Low	Forwarded to eSPI	Yes.

continued...

I/O Address	Read Target	Write Target	Internal Unit (Unless[E]: External)[2]	Separate Enable/Disable
				ESPI_CS1IORE.LGE
208-20Fh	Gameport High	Gameport High	Forwarded to eSPI	Yes ESPI_CS1IORE.HGRE

Notes: 1. Only if the Port 61 Alias Enable bit (GIC.P61AE) bit is set. Otherwise, the cycle is internally terminated by the PCH.
2. Destination of eSPI when eSPI Disabled pin strap is 0.
3. This includes byte, word or double-word (DW) access at I/O address 80h.
4. Read to Port 70 will read from RTC in PCH-S. Write to Port 70h will generate a write to ITSS (in SOC-S), and a write to RTC (in PCH-S).

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 15. Variable I/O Decode Ranges

Range Name[1]	Mappable	Size (Bytes)	Target
IDE Bus Host	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
eSPI Generic 1	Anywhere in 64K I/O Space	4 to 256 Bytes	eSPI
eSPI Generic 2	Anywhere in 64K I/O Space	4 to 256 Bytes	eSPI
eSPI Generic 3	Anywhere in 64K I/O Space	4 to 256 Bytes	eSPI
eSPI Generic 4	Anywhere in 64K I/O Space	4 to 256 Bytes	eSPI
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 1-24
Keyboard and Text (KT)	Anywhere in 64K I/O Space	8	Intel® AMT Keyboard and Text
DMI General Purpose I/O Ranges (1 to 3)	Anywhere in 64K I/O Space	4 to 256 Bytes	General purpose

Note: 1. All ranges are decoded directly from IOC.

3.2 Memory Map

The following table shows (from the processor perspective) the memory ranges that the PCH will decode. Cycles that are not directed to any of the internal memory targets, will be host aborted.

PCIe cycles generated by external PCIe hosts 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). Software must not attempt locks to the PCH memory-mapped I/O ranges.

NOTE

Total ports are different for the different SKUs.

Table 16. 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
FEF0 0000 - FEFF FFFF	eSPI or SPI	uCode Patch Region Enable UCPR.UPRE is set
FFC0 0000 - FFC7 FFFF FF80 0000 - FF87 FFFF	eSPI or SPI	Bit 8 in BIOS Decode Enable Register is set
FFC8 0000 – FFCF FFFF FF88 0000 - FF8F FFFF	eSPI or SPI	Bit 9 in BIOS Decode Enable Register is set
FFD0 0000 - FFD7 FFFF FF90 0000 - FF97 FFFF	eSPI or SPI	Bit 10 in BIOS Decode Enable Register is set
FFD8 0000 – FFDF FFFF FF98 0000 - FF9F FFFF	eSPI or SPI	Bit 11 in BIOS Decode Enable Register is set
FFE0 0000 - FFE7 FFFF FFA0 0000 - FFA7 FFFF	eSPI or SPI	Bit 12 in BIOS Decode Enable Register is set
FFE8 0000 – FFEF FFFF FFA8 0000 – FFAF FFFF	eSPI or SPI	Bit 13 in BIOS Decode Enable Register is set
FFF0 0000 - FFF7 FFFF FFB0 0000 - FFB7 FFFF	eSPI or SPI	Bit 14 in BIOS Decode Enable Register is set
FFFC 0000 - FFFF FFFF	eSPI, SPI	Always enabled.
FFF8 0000 - FFFB FFFF FFB8 0000 - FFBF FFFF	eSPI or SPI	Always enabled.
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
FED4 0000 - FED4 7FFF	SPI (set by strap)	TPM and Trusted Mobile KBC

continued...

Memory Range	Target	Dependency/Comments
128 KB anywhere in 4 GB range	LAN Controller (CSR registers)	Enable via standard PCI mechanism (Device 31:Function 6)
4 KB anywhere in 4 GB range	LAN Controller (LAN space on Flash)	Enable via standard PCI mechanism (Device 31:Function 6)
64 KB anywhere in 64-bit address range	USB 3.2 Host Controller	Enable via standard PCI mechanism (Device 20, Function 0)
2 MB anywhere in 4 GB range	USB Device Controller	Enable via standard PCI mechanism (Device 20, Function 1)
24 KB anywhere in 4 GB range	USB Device Controller	Enable via standard PCI mechanism (Device 20, Function 1)
16 KB anywhere in 64-bit addressing space	Intel® HD Audio Subsystem	Enable via standard PCI mechanism (Device 31, Function 3)
4 KB anywhere in 64-bit addressing space	Intel® HD Audio Subsystem	Enable via standard PCI mechanism (Device 31, Function 3)
64 KB anywhere in 64-bit addressing space	Intel® HD Audio Subsystem	Enable via standard PCI mechanism (Device 31, Function 3)
1MB anywhere in 4 GB range	eSPI	LPC Generic Memory Range. Enable via setting bit<wbr>[0] of the LPC Generic Memory Range register (D31:F0:offset 98h).
64 KB anywhere in 4 GB range	eSPI	eSPI Generic Memory Range (for the 2nd, 3rd and 4th eSPI CS#). Enable bit via setting bit<wbr>[0] of the respective Memory Range register (D31:F0)
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 1-24	Enable via standard PCI mechanism
Prefetchable Memory Base/Limit anywhere in 64-bit address range	PCI Express* Root Ports 1-24	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
12 4 KB slots anywhere in 64-bit address range	I^3C function has 8 KB BAR, all others (I^2C/SPI/UART) are 4 KB.	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

continued...

Memory Range	Target	Dependency/Comments
Two 32 KB anywhere in 64-bit address range	THC #0, #1	Enable via standard PCI mechanism
DMI General Purpose Memory ranges (1 to 3)	General purpose	Enable via setting the Decode Enable bit of the respective Memory Range register
MMIO resources for VMD managed devices	Storage devices	Enable through the VMD device in PCH

4.0 Audio Voice and Speech

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 processor 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 the PCH 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

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.
I^2S	Inter IC Sound. A serial bus using PCM.
MEMs	Micro electrical mechanical Systems. For AVS devices such as Digital MEMs Microphones.
MSI	Message Signaled Interrupt. An in-band method of signaling an interrupt.
PCM	Pulse Code Modulation. Modulation with amplitude coded into stream.
PDM	Pulse Density Modulation. Modulation with amplitude coded by pulse density.
SDI	Serial Data In.
SDO	Serial Data Out.
VOIP	Voice Over Internet Protocol

Table 18. References

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

4.1 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. With the converged audio architecture initiatives, it is also the baseline audio host controller for phone and tablet platforms with optional DSP support. Intel® HD Audio controller capabilities are listed as follows:

- PCI / PCI Express* controller
  - Option to hide PCI configuration space and use ACPI method for enumeration
- Supports data transfers, descriptor fetches, and DMA position writes using VC0 or VC1
- Independent Bus Host logic for 19 general purpose DMA streams: 10 input and 9 output
- Supports variable length stream slots
- Each general purpose stream supports up to:
  - 16 channels per stream
  - 32 bits/sample
  - 192 kHz sample rate
- Supports memory-based command/response transport
- Supports optional Immediate Command/Response mechanism
- Supports input and output stream synchronization
- Supports MSI interrupt delivery
- Support for ACPI D3 and D0 Device States
- Supports Function Level Reset (FLR)
  - Only if exposed as a PCI Express device (or ACPI method)
- Support Converged Platform Power Management (CPPM)
  - Support 1 ms of buffering with all DMA running with maximum bandwidth

4.2 Audio DSP Capabilities

The Audio DSP offload engine is a feature providing low power DSP functionality and offloads the audio processing operation from the host Processor. 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:

- Up to 2 x 400 MHz Tensilica* LX7 HIFI4 DSP Cores
- Up to 2816 KB of L2 HP SRAM by all DSP Core

- L2 uncache memory accessing up to 16 x 16 MB of remote DDR region
- Up to 3 x 8 ch GPDMA for data transfers to and from DSP I/O peripherals
- DSP offload for low power audio rendering and recording
- Various DSP functions provided by Tensillica Core: MP3, AAC, Dolby Digital*, etc.
- Host downloadable DSP FW functions
- Voice call processing enhancement
- HW based DSP accelerators, for example, Machine Learning block and SHA engine

4.3 Intel® High Definition Audio Interface Capabilities

The Intel® HD Audio interface is a 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:

- The SDI signals to support 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 3.3 V and 1.8 V.

4.4 Direct Attached Digital Microphone (PDM) Interface

The direct attached digital microphone interface is a 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:

- Up to 2 Digital Mic Ports with to 2 Digital Mic Modules per Digital Mic Port
- Ability to combine multiple Digital Mic Ports to for mic arrays that are synchronized on sampling rate basis
- 2 PCM Audio Streams with independent PCM sampling rates per Digital Mic Port
- Ability to map each Digital Mic Port stereo PCM streams output to a sub-set of a multi-channel PCM stream data transferred to an Audio Link Hub
  - Support dynamic scaling up/down of microphone channels array after the stream has started
- Support child clock input mode of operation

4.5 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

4.6 I²S / PCM Interface

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

- Up to 3 bi-directional I²S / PCM ports to support up to 16 channels per port
- Controller/device mode support for run-time selection
- Each I²S / PCM ports are able to support multiple devices using PCM mode (also known as TDM Mode)
- Support multi I²S / PCM port synchronization

4.7 Intel® Display Audio Interface

The Intel® iDisp Audio link is a feature offering connection to the Intel® iDisp Audio codec. The Intel® iDisp Audio codec is used to provide audio streams routing to the integrated HDMI and DP links, through the existing Intel® HD Audio controller SW stacks. This iDisp audio codec used to be attached to the Intel® HD Audio link, however, it transitioned to a dedicated 3-wire iDisp Audio link to save pin counts on the compute tile, as well as providing finer grain power management to the audio link interfaces. The Intel® iDisp Audio codec is enumerated by the Intel® HD Audio driver software. Features for the Intel® HD Audio interface is provided below:

- 1 SDI signal to support 1 iDisp audio codec
- Drives 96 MHz frequency BCLK support
  - SDO single pumped to 96 Mb/s

  - — SDI single pumped up to 96 Mb/s
- Provides cadence for 44.1 kHz-based sample rate output.

4.8 MIPI® SoundWire* Interface

The SoundWire interface is a 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 and PDM streaming capabilities. SoundWire device class initiative for audio is bringing standardization to the audio codec SW stack. These devices are enumerated based on vendor / device ID of the SoundWire device reporting, allowing vendor customization of audio codec SW if desired. SoundWire interface capabilities are listed as follows:

- Up to 4 SoundWire interfaces frame rate synchronized on global periodic events
- SNDW[0] supports 4 data lane per SoundWire interface
- Support SoundWire Device Class Specification for Audio Controls and Memories
- Up to 8 PCM bidirectional streams per SoundWire interface
  - — Direction is programmable as either input or output stream
- Up to 8 channels per PCM streams
- Interrupt / PME wake capable on DATA pin assertion in low power state

NOTE

PDM support exists in design but is not enabled for SNDW links.

4.9 Signal Description

Signal Name	Type	Description
Intel® High Definition Audio Signals		
GPP_R04/**HDA_RST#**	O	**Intel HD Audio Reset**: Host H/W reset to internal and external codecs.
GPP_R01/**HDA_SYNC**/ I2S0_SFRM/DMIC_CLKB1	O	**Intel HD Audio Sync**: 48 kHz fixed rate frame sync to the codecs.
GPP_R00/**HDA_BCLK**/ I2S0_SCLK/HDAPROC_BCLK/ DMIC_CLKA1	O	**Intel HD Audio Bit Clock**: Up to 24 MHz serial data clock generated by the Intel® HD Audio controller.
GPP_R02/**HDA_SDO**/I2S0_TXD/ HDAPROC_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_R03/**HDA_SDI0**/I2S0_RXD/ HDAPROC_SDI	I/O	**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_R05/**HDA_SDI1**/I2S1_RXD	I/O	**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_J14/**HDAPROC_BCLK** or GPP_R00/HDA_BCLK/I2S0_SCLK/ **HDAPROC_BCLK**/DMIC_CLKA1	O	**Intel® Display Audio Bit Clock**: Serial data clock generated by the Intel HD Audio controller. the clock output frequency is up to 192 Mb/s.
GPP_J13/**HDAPROC_SDO** or GPP_R02/HDA_SDO/I2S0_TXD/ **HDAPROC_SDO**	O	**Intel® Display Audio Serial Data Out**: Serial TDM data output to the codec. PCH supports data rate of up to 192 Mb/s.
GPP_J12/**HDAPROC_SDI** or GPP_R03/HDA_SDI0/I2S0_RXD/ **HDAPROC_SDI**	I	**Intel® Display Audio Serial Data In**: Serial TDM data input from the codec. PCH supports data rate of up to 192 Mb/s.
I²S / PCM Interface		
GPP_R00/HDA_BCLK/ **I2S0_SCLK**/HDAPROC_BCLK/ DMIC_CLKA1	I/O	**I²S / PCM serial bit clock 0**: Serial bit clock used to control the timing of a transfer. Can be generated internally (Host mode) or taken from an external source (Device mode).
GPP_R08/**I2S1_SCLK**	I/O	**I²S / PCM serial bit clock 1**:Serial bit clock is used to control the timing of a transfer. Can be generated internally (Host mode) or taken from an external source (Device mode).
GPP_D08/**I2S2_SCLK**/ DMIC_CLK_A0	I/O	**I²S / PCM serial bit clock 2**: Serial bit clock is used to control the timing of a transfer. Can be generated internally (Host mode) or taken from an external source (Device mode).
GPP_R01/HDA_SYNC/ **I2S0_SFRM**/DMIC_CLKB1	I/O	**I²S / PCM serial frame indicator 0**: This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Host mode) or taken from an external source (Device mode).
GPP_R07/**I2S1_SFRM**	I/O	**I²S / PCM serial frame indicator 1**: This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Host mode) or taken from an external source (Device mode).
GPP_D05/**I2S2_SFRM**/ CNV_RF_RESET#/DMIC_DATA0	I/O	**I²S / PCM serial frame indicator 1**: This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Host mode) or taken from an external source (Device mode).
GPP_R02/HDA_SDO/**I2S0_TXD**/ HDAPROC_SDO	O	**I²S / PCM transmit data (serial data out)0**: Serial data out line. Sample length is a function of the selected serial data sample size.
GPP_R06/**I2S1_TXD**	O	**I²S / PCM transmit data (serial data out)1**: Serial data out line. Sample length is a function of the selected serial data sample size.
GPP_D06/**I2S2_TXD**/ CRF_CLKREQ/DMIC_CLK_B0	O	**I²S / PCM transmit data (serial data out)1**: Serial data out line. Sample length is a function of the selected serial data sample size.

continued...

Signal Name	Type	Description
GPP_R03/HDA_SDI0/**I2S0_RXD**/ HDAPROC_SDI	I	**I²S / PCM receive data (serial data in)0**: Serial data in line. Sample length is a function of the selected serial data sample size.
GPP_R05/HDA_SDI1/**I2S1_RXD**	I	**I²S / PCM receive data (serial data in)1**: Serial data in line. Sample length is a function of the selected serial data sample size.
GPP_D07/**I2S2_RXD**/ DMIC_DATA1	I	**I²S / PCM receive data (serial data in)1**: Serial data in line. Sample length is a function of the selected serial data sample size.
GPP_B11/**I2S_MCLK**	O	**I²S / PCM Host reference clock 0**: This signal is the host reference clock that connects to an audio codec.
DMIC Interface		
GPP_D08/I2S2_SCLK/ **DMIC_CLK_A0** or GPP_S02/SNDW1_CLK/ **DMIC_CLK_A0**/SNDW0_DATA1	O	**Digital Mic Clock A0**:Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz. Duplication for clock pin (instance A) in case platform wanted to separate clock connection for left channel mic vs right channel mic. For the case of sharing single clock connection to both left and right channel mics, clock pin (instance A) should be used.
GPP_S06/SNDW3_CLK/ **DMIC_CLK_A1** or GPP_R00/HDA_BCLK/I2S0_SCLK/ HDAPROC_BCLK/**DMIC_CLKA1**	O	**Digital Mic Clock A1**:Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz. Duplication for clock pin (instance A) in case platform wanted to separate clock connection for left channel mic vs right channel mic. For the case of sharing single clock connection to both left and right channel mics, clock pin (instance A) should be used.
GPP_S04/SNDW2_CLK/ **DMIC_CLK_B0**/SNDW0_DATA3 or GPP_D06/I2S2_TXD/ CRF_CLKREQ/**DMIC_CLK_B0**	O	**Digital Mic Clock B0**:Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz. Duplication for clock pin (instance B) in case platform wanted to separate clock connection for left channel mic vs right channel mic. For the case of sharing single clock connection to both left and right channel mics, clock pin (instance B) can be disconnected.
or GPP_S05/SNDW2_DATA/ **DMIC_CLK_B1** GPP_R01/ 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. Duplication for clock pin (instance B) in case platform wanted to separate clock connection for left channel mic vs right channel mic. For the case of sharing single clock connection to both left and right channel mics, clock pin (instance B) can be disconnected.
GPP_S03/SNDW1_DATA/ **DMIC_DATA0**/SNDW0_DATA2 or GPP_D05/I2S2_SFRM/ CNV_RF_RESET#/**DMIC_DATA0**	I	**Digital Mic Data**:Serial data input from the digital mic.
GPP_S07/SNDW3_DATA/ **DMIC_DATA1** or GPP_D07/I2S2_RXD/ **DMIC_DATA1**	I	**Digital Mic Data**:Serial data input from the digital mic.
SoundWire Interface		

continued...

Signal Name	Type	Description
GPP_S00/**SNDW0_CLK**	I/O	**SoundWire Clock**: Serial bit clock used to control the timing of a transfer.
GPP_S01/**SNDW0_DATA0**	I/O	**SoundWire Data**: Serialized data line containing framing and data being transmitted/received.
GPP_S02/**SNDW1_CLK**/ DMIC_CLK_A0/**SNDW0_DATA1**	I/O	**SoundWire Clock**: Serial bit clock used to control the timing of a transfer. **SoundWire Data**: Serialized data line containing framing and data being transmitted/ received.
GPP_S03/**SNDW1_DATA**/ DMIC_DATA0/**SNDW0_DATA2**	I/O	**SoundWire Data**: Serialized data line containing framing and data being transmitted/received.
GPP_S04/**SNDW2_CLK**/ DMIC_CLK_B0/**SNDW0_DATA3**	I/O	**SoundWire Clock**: Serial bit clock used to control the timing of a transfer. **SoundWire Data**: Serialized data line containing framing and data being transmitted / received.
GPP_S05/**SNDW2_DATA**/ DMIC_CLK_B1	I/O	**SoundWire Data**: Serialized data line containing framing and data being transmitted / received.
GPP_S06/**SNDW3_CLK**/ DMIC_CLK_A1	I/O	**SoundWire Clock**: Serial bit clock used to control the timing of a transfer.
GPP_S07/**SNDW3_DATA**/ DMIC_DATA1	I/O	**SoundWire Data**: Serialized data line containing framing and data being transmitted / received.
SNDW_RCOMP	A	SoundWire Resistor compensation.

4.10 Integrated Pull-Ups and Pull-Downs

Table 19. Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type	Value
HDA_SYNC	Pull-down	20 kohm
HDA_SDO	Pull-down	20 kohm
HDA_SDI[1:0]	Pull-down	20 kohm
DMIC_DATA[1:0]	Pull-down	20 kohm
SNDW0_DATA[3:0]	Pull-down	5 kohm
SNDW[3:1]_DATA	Pull-down	5 kohm

4.11 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
High Definition Audio Interface				
HDA_RST#	Primary	Asserted	Asserted	Asserted
HDA_SYNC	Primary	Internal Pull-down	Driven Low	Internal Pull-down
HDA_BCLK	Primary	Driven Low	Driven Low	Driven Low
HDA_SDO	Primary	Internal Pull-down	Driven Low	Internal Pull-down

continued...

Signal Name	Power Plane	During Reset[2]	Immediately After Reset[2]	S4/S5
HDA_SDI[1:0]	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down
I²S/PCM Interface				
I2S[2:0]_SCLK	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down
I2S[2:0]_SFRM	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down
I2S[2:0]_TXD	Primary	Driven Low	Driven Low	Driven Low
I2S[2:0]_RXD	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down
I2S_MCLK	Primary	Driven Low	Driven Low	Driven Low
Display Audio Interface				
HDAPROC_BCLK	Primary	Driven Low	Driven Low	Internal Pull-down
HDAPROC_SDO	Primary	Internal Pull-down	Driven Low	Internal Pull-down
HDAPROC_SDI	Primary	Internal Pull-down	Internal Pull-down	Driven Low
DMIC Interface				
DMIC_CLK_A[1:0]	Primary	Driven Low	Driven Low	Driven Low
DMIC_CLK_B[1:0]	Primary	Driven Low	Driven Low	Driven Low
DMIC_DATA[1:0]	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down
SoundWire Interface				
SNDW0_DATA[3:0]	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down
SNDW[3:1]_DATA	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down
SNDW[3:0]_CLK	Primary	Driven Low	Driven Low	Driven Low
Note: 1. Pull-down is enabled with PLTRST# is asserted for the following signals: HDA_SYNC, HDA_SDO, HDAPROC_SDO				

5.0 Ballout Definition

For information on ballout, download the pdf, click on the navigation pane and refer the spreadsheet, **833778-001_Ballout.xlsx**.

6.0 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, which is embedded in the PCH.

The Integrated Connectivity (CNVi) solution consists of the following entities:

- The containing chip (the PCH which contains the Connectivity Controller)
- Buttress (as applicable to each platform, and coupled the Connectivity Controller)
- 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 21. Acronyms

Acronyms	Description
BRI	Bluetooth* Radio Interface
CNVi	Connectivity Integrated
SCU	System Controller Unit . It is the controller unit of CNVi.
RGI	Radio Generic interface
IP	Literally, Intellectual Property. IP refers to architecture, design, validation, and software components collectively delivered to enable one or more specific to the PCH features.
MFUART	Multifunction Universal Asynchronous Receiver/Transmitter
UART	Universal Asynchronous Receiver/Transmitter

Table 22. 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

6.1 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.

6.2 Signal Description

Signal Name	Type	Description
GPIO fixed functions (Signals for Integrated Connectivity (CNVi) and Discrete Connectivity (CNVd) functions		
GPP_D08/**I2S2_SCLK**/DMIC_CLK_A0	I/O	For CNVi: Unused For discrete connectivity with UART host support: Optional Bluetooth* I^2S bus clock
GPP_D05/I2S2_SFRM/**CNV_RF_RESET#**/ DMIC_DATA0	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.
GPP_D06/**I2S2_TXD**/CRF_CLKREQ/ DMIC_CLK_B0	O	For CNVi: Unused For discrete connectivity with UART host Bluetooth* support: Optional Bluetooth* I^2S bus data output (input to Bluetooth* module)
GPP_D07/**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_J01/**CNV_BRI_DT**/UART0_RTS#	O	For CNVi: BRI bus TX. For discrete connectivity with UART host support: Bluetooth* UART RTS#
GPP_J02/**CNV_BRI_RSP**/UART0_RXD	I	For CNVi: BRI bus RX. For discrete connectivity with UART host support: Bluetooth* UART RXD

continued...

Signal Name	Type	Description
GPP_J03/**CNV_RGI_DT**	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_J04/**CNV_RGI_RSP**/UART0_CTS#	I	For CNVi: RGI bus RX. For discrete connectivity with UART host support: Bluetooth* UART CTS#
GPP_D06/I2S2_TXD/**CRF_CLKREQ**/ DMIC_CLK_B0	O	For CNVi: PCH to CRF wake indication
GPP_J00/**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_J05/**CNV_MFUART2_RXD**	I	For CNVi and discrete connectivity: Optional WLAN/Bluetooth* WWAN co-existence signal (Input)
GPP_J06/**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		
USB2P_10	I/O	Bluetooth* USB host bus (positive) for discrete connectivity. Optional to connect to a Bluetooth* USB+ pin on the Bluetooth* module. 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. Other USB 2.0 ports can be selected for this function.
PCIE_12_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. Other PCIe* ports can be selected for this function.
PCIE_12_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. Other PCIe* ports can be selected for this function.
PCIE_12_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. Other PCIe* ports can be selected for this function.

continued...

Signal Name	Type	Description
PCIE_12_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. 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. 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. Other PCIe* clocks can be selected for this function.
CL_RST#	O	Wi-Fi* CLINK host bus reset for discrete connectivity with CLINK support. Optional to connect to a Wi-Fi* CLINK reset pin on the Wi-Fi* module.
CL_DATA	I/O	Wi-Fi* CLINK host bus data for discrete connectivity with CLINK support. Optional to connect to a Wi-Fi* CLINK data pin on the Wi-Fi* module.
CL_CLK	O	Wi-Fi* CLINK host bus clock for discrete connectivity with CLINK support. Optional to connect to a Wi-Fi* CLINK clock pin on the 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 PCH 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) *Note:* Signal name not available in PCH ballmap. This is a representation of GPIO used as CNVi signal.
W_Disable2# (GPIO)	O	Used for Bluetooth* RF-Kill control. This pin can be connected to a platform switch or to PCH 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) *Note:* Signal name not available in PCH ballmap. This is a representation of GPIO used as CNVi signal.

6.3 Integrated Pull-ups and Pull-downs

Signal	Resistor	Value	Notes
CNV_BRI_RSP	Pull up	20 kohm	
CNV_RGI_RSP	Pull up	20 kohm	

6.4 I/O Signal Planes and States

Signal Name	Power plane	During Reset[1]	Immediately After Reset[1]	S4/S5
CNV_RF_RESET#	Primary	Undriven	Driven Low	Undriven
CRF_CLKREQ	Primary	Driven High	Driven High	Driven High
CNV_MFUART2_RXD	Primary	Undriven	Undriven	Undriven
CNV_MFUART2_TXD	Primary	Undriven	Undriven	Undriven
CNV_BRI_DT	Primary	Driven High	Driven High	Driven High

continued...

Signal Name	Power plane	During Reset[1]	Immediately After Reset[1]	S4/S5
CNV_BRI_RSP	Primary	Powered (input, PU)	Powered (input, PU)	Powered (input, PU)
CNV_RGI_DT	Primary	Driven High	Driven High	Driven High
CNV_RGI_RSP	Primary	Powered (input, PU)	Powered (input, PU)	Powered (input, PU)
CNV_WT_CLKP	Primary	Undriven	Driven Low	Undriven
CNV_WT_CLKN	Primary	Undriven	Driven Low	Undriven
CNV_WT_D0P	Primary	Undriven	Driven Low	Driven High
CNV_WT_D0N	Primary	Undriven	Driven Low	Driven High
CNV_WT_D1P	Primary	Undriven	Driven Low	Driven High
CNV_WT_D1N	Primary	Undriven	Driven Low	Driven High
CNV_WR_CLKP	Primary	Undriven	Undriven	Powered (input)
CNV_WR_CLKN	Primary	Undriven	Undriven	Powered (input)
CNV_WR_D0P	Primary	Undriven	Undriven	Powered (input)
CNV_WR_D0N	Primary	Undriven	Undriven	Powered (input)
CNV_WR_D1P	Primary	Undriven	Undriven	Powered (input)
CNV_WR_D1N	Primary	Undriven	Undriven	Powered (input)
Note: 1. Reset reference for primary well pins is PCH_RSMRST#.				

7.0 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 the Controller Link interface. The Controller Link clock frequency is 30.0 MHz. The Controller Link interface voltage supported is 1.25 V nominal.

NOTE

Refer to WNIC product datasheets for supported data rate and clock.

Table 23. Acronyms

Acronyms	Description
CL	Controller Link
WLAN	Wireless Local Area Network
WNIC	Wireless Network Interface Card

7.1 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value (Ohm)	Notes
CL_DATA	Pull-up Pull-down	31.25 100	I/O Signal Planes and States on page 49
CL_CLK	Pull-up Pull-down	31.25 100	

7.2 I/O Signal Planes and States

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

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 PCH_RSMRST#.

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

The WLAN has transitioned to 1.8 V for external CL_RST# pin and the PCH controller Link I/O buffer drives 1.8 V only on this pin.

7.4 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	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.

8.0 Direct Enhanced Serial Peripheral Interface (Direct eSPI)

The Direct eSPI interface is used in order to support the secure eSPI link between the processor and the PCH. The processor is a Direct eSPI controller and PCH is a Direct eSPI target. The Direct eSPI interface features as below.

- 1.8 V support only.
- Compliant to eSPI base specification 1.0.
- Supports 50 MHz Single/Dual/Quad modes.
- The Direct eSPI interface have a return clock eSPI_RCLK. In order to improve the eSPI bandwidth between the processor and the PCH, an eSPI return clock is used to increase the eSPI clock frequency from 50 MHz to 100 MHz. The PCH loopbacks the eSPI clock and returns the eSPI_RCLK to the processor. The PCH launches the eSPI response on the new return clock eSPI_RCLK.

Table 24. Acronyms

Acronyms	Description
OOB	Out-of-Band
TAR	Turn-around cycle
Intel® CSE	Intel® Converged Security Engine
VW	Virtual Wire

Table 25. References

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

8.1 Functional Description

8.1.1 Configuration Flow

The processor's present pin strap is tied with the PCH eSPI enable strap pin during the platform configuration flow. Depending on the processor's present pin strap status and PCH eSPI enable pin strap status, the PCH Direct eSPI target takes the decision to configure the flash, VW, and peripheral channel.

In an instance when an EC is attached to the PCH eSPI controller, the PCH Direct eSPI target waits if the PCH eSPI controller's flash, VW, and the peripheral channel are ready before it can set the ready bit of the respective channel on Direct eSPI interface.

The PCH Direct eSPI target configures the channel depending on the GET_CONFIG/ SET_CONFIG/GET_STATUS/INBAND Reset commands and the latest channel/link capability information (e.g. flash, RPMC eSPI capability or SPI flash SFDP), which were gathered from its downstream PCH eSPI EC/other devices or SPI controller.

8.1.2 PCH Direct eSPI Registers

The PCH Direct eSPI target maintains the standard eSPI registers defined in the eSPI specification including SAFS and RPMC registers in offset, 0x40/0x44. The Channel 0 Capabilities and Configurations is reset by the Platform Reset (PLTRST#). All the other registers are reset by ESPI_RESET#.

Refer to Intel® 800 Series Chipset Family Platform Controller Hub Datasheet, Volume 2 of 2 (#834576) for more details on PCH Direct eSPI.

8.1.3 PCH Direct eSPI Reset Handling

When the PCH eSPI controller ESPI_RESET# asserts, the PCH Direct eSPI target resets its entire eSPI interface-related logic. Before ESPI_RESET# assertion, the processor issues a PLTRST# assertion VW over the Direct eSPI interface. The PCH Direct eSPI target uses this VW for reset isolation from the rest of the eSPI controller, which is available on the early boot side reset domain. Once isolation is completed, the PCH Direct eSPI target forwards PLTRST# assertion VW to PCH PMC. The ESPI_RESET# assertion occurs after the PLTRST# assertion VW received by the PCH.

8.1.4 PCH Direct eSPI Peripheral Channel Support

The PCH Direct eSPI target forwards the received peripheral channel IO/memory cycles from the processor Direct eSPI, PCH eSPI, or PCH internal from/to the upper layer eSPI switch. The eSPI switch then decodes the address and forwards it to the corresponding destination (Ex. EC, other IP) accordingly.

The PCH Direct eSPI target defers the downstream non-posted cycle and sends back the completion. The order of the Posted/Non-Posted cycles in both directions enforced to follow the eSPI ordering rule in the eSPI base spec "5.4 Transaction Ordering Rule".

The peripheral channel is enabled automatically after PLTRST# deassertion (tunneled as a VW). The peripheral channel is enabled before any downstream transactions are delivered. The PCH Direct eSPI target will not send upstream bus controlling requests until the Direct eSPI BME bit has been set..

In preparation for host reset, the processor Direct eSPI controller completes all downstream peripheral channel (P/Ch.) pending transactions and quiesces all its P/Ch request queues before sending the VW Host_Reset_Warn to the PCH Direct eSPI target. The PCH Direct eSPI target completes all P/Ch pending transactions (including downstream completions) and quiesces all upstream P/Ch request queues before sending Host_Reset_Ack VW back to the processor Direct eSPI controller. The PCH Direct eSPI target must continue to be able to receive downstream posted transactions until the channel is disabled with the assertion of PLTRST# (tunneled as VW). After sending the HOST_RST_ACK, the PCH Direct eSPI target does not send any peripheral channel transaction.

The PCH Direct eSPI controller automatically clears the P/Ch enable bit and BME bit upon PLTRST# assertion (tunneled as VW) set by BIOS.

A host reset Warn/Ack handshake (using VWs) will always accompany the assertion of PLTRST# (tunneled as a VW). The PCH Direct eSPI target Status register's P/Ch bits are cleared and locked after sending the Host_Reset_Ack VW. The host re-enables the P/Ch.

An unplanned reset event in the processor or the PCH might cause an abrupt assertion of PLTRST# and ESPI_RESET#, abruptly terminating any pending requests/completions, without a reset Warn/Ack handshake. In this scenario, the PLTRST# assertion is not tunneled as VW. The Direct eSPI interface, all channels, and all VWs revert back to their default states due to the simultaneous assertion of ESPI_RESET#.

8.1.5 PCH Direct eSPI VW Channel Support

The PCH Direct eSPI target supports Virtual Wire channel in both transmit and receive directions. It receives VWs from the internal eSPI switch or from the processor Direct eSPI controller.

The processor Direct eSPI enables the VW channel after ESPI_RESET# has been de-asserted and the eSPI link has been initialized and configured. The VW channel is automatically disabled upon ESPI_RESET# assertion.

The PCH Direct eSPI target must not send any VW (except Device_Boot_Load_done/status) until the processor Direct eSPI controller has sent its first set of VWs to the PCH. Otherwise, this might lead to unexpected system behavior due to mismatched VW states between the PCH and the processor.

The PCH eSPI target does not generate VW cycles internally. All VWs including device_boot_load_done/status VW from/to EC are forwarded from/to eSPI switch and sent to/from the processor Direct eSPI controller.

The PCH Direct eSPI target resets VW states based on the eSPI base specification. After sending the HOST_RST_ACK, the PCH Direct eSPI target does not send any host domain VWs such as SMI#, SCI#, RCIN#, or IRQ. Until PLTRST# is de-asserted, host domain VWs are not sent from the Direct eSPI controller or target.

The PCH Direct eSPI target accepts the VW packets at any time as long as the channel is enabled.

8.1.6 PCH Direct eSPI OOB Channel Support

The PCH Direct eSPI target's OOB channel transmits and receives HW/FW OOB packets from the processor Direct eSPI controller. It forwards all received OOB packets between the PCH eSPI switch and the processor Direct eSPI controller.

The OOB channel is automatically disabled upon ESPI_RESET# assertion. An unplanned reset event in the processor or the PCH may cause abrupt PCH_RSMRST# assertion without a reset/warn handshake. It causes the channel to be disabled for both the Direct eSPI controller and target, abruptly terminating any pending requests/completions.

8.1.7 PCH Direct eSPI Flash Channel Support

The PCH Direct eSPI device target supports SAF mode, including RPMC. For RPMC support, the PCH Direct eSPI parameters in the SAF capability and configuration registers are sourced from either SPI Flash's RPMC SFDP table or EC's SAF RPMC register depending on the PCH's flash sharing mode.

The PCH eSPI controller reads the "Device 0" SAF registers (0x40 and 0x44) and saves them inside the controller in the private registers. The PCH SPI controller reads the SFDP RPMC register and saves it inside the controller in the private registers.

The PCH Direct eSPI target uses either of these two sets of parameters depending on the MAF or SAF mode and initializes the device's SAF 0x40/0x44 register.

Only one RPMC flash device is supported on device 0 in SAF mode. In SAF mode, the PCH Direct eSPI target defers the SAF cycles from the connected processor Direct eSPI controller. It forwards the SAF requests to the PCH eSPI host controller. When it receives the completion back from the PCH eSPI host controller, the PCH Direct eSPI target asserts an interrupt/flash_c_avail to send the completion back to the processor.

8.1.8 PCH eSPI Switch

The eSPI switch decodes and routes the eSPI cycles among the following agents:

Peripheral Channel:

- PCH Direct eSPI controller
  - EC/PCH eSPI device IO, upstream/downstream memory cycles.
- Processor Direct eSPI Controller (to/from 1-4 eSPI devices, including EC)
  - EC/PCH eSPI device IO, upstream/downstream memory/completion cycles.
- PCH Peripheral Channel Controller
  - PCH RTC IO cycles.
- PCH eSPI controller
  - SPI/eSPI host/device/switch config, SPI/eSPI Config/MMIO and private registers.

Virtual Wire Channel:

- PCH Direct eSPI controller
  - Processor PMC
  - Processor ITSS
  - Processor GPIO
  - PCH eSPI
- Processor Direct eSPI Controller (to/from 1-4 eSPI devices, including EC)
- PCH eSPI Virtual Wire Channel Controller
  - PCH GPIO/PMC/eSPI
- PCH eSPI controller
  - DnX, host reset warn/ack, OOB reset warn ack, Device_Boot_ Load_Done/ Status; Error_Fatal / Error_NonFatal/NMI/Aggregated IRQ

OOB Channel:

- PCH Direct eSPI controller
  - Processor PMC: OOB messages for EC (e.g. PECI).
  - Processor CSE: Primary OOB messages.
  - Processor Silicon Security Engine: Primary OOB messages.
- Processor Direct eSPI Controller (to/from 1-4 eSPI devices, including EC)
- PCH OOB Channel Controller

  - PCH CSE: Primary OOB messages.
  - PCH Silicon Security Engine: Primary OOB messages
  - PCH PMC: Primary OOB messages.
- PCH eSPI controller
  - PCH temperature/RTC OOB messages.

Flash Channel:

- PCH Direct eSPI controller
  - Processor BIOS: through PCH SPI controller, including both flash and RPMC
  - Processor CSE: through PCH SPI controller, including both flash and RPMC
  - Processor Silicon Security Engine: through PCH SPI controller
  - PCH SPI controller: Processor descriptor access/recovery
- Processor Direct eSPI Controller (Device0/EC)
- PCH Flash Channel Controller
  - PCH CSE: through PCH SPI controller, including both flash and RPMC
  - PCH Silicon Security Engine: through PCH SPI controller
  - PCH GbE: through PCH SPI controller
- PCH SPI controller
  - PCH descriptor access/recovery

8.1.9 Flash Descriptor Address Swapping

The processor and the PCH share the same flash on PCH side either in SAF or MAF mode. Both components require its own version of the descriptors.

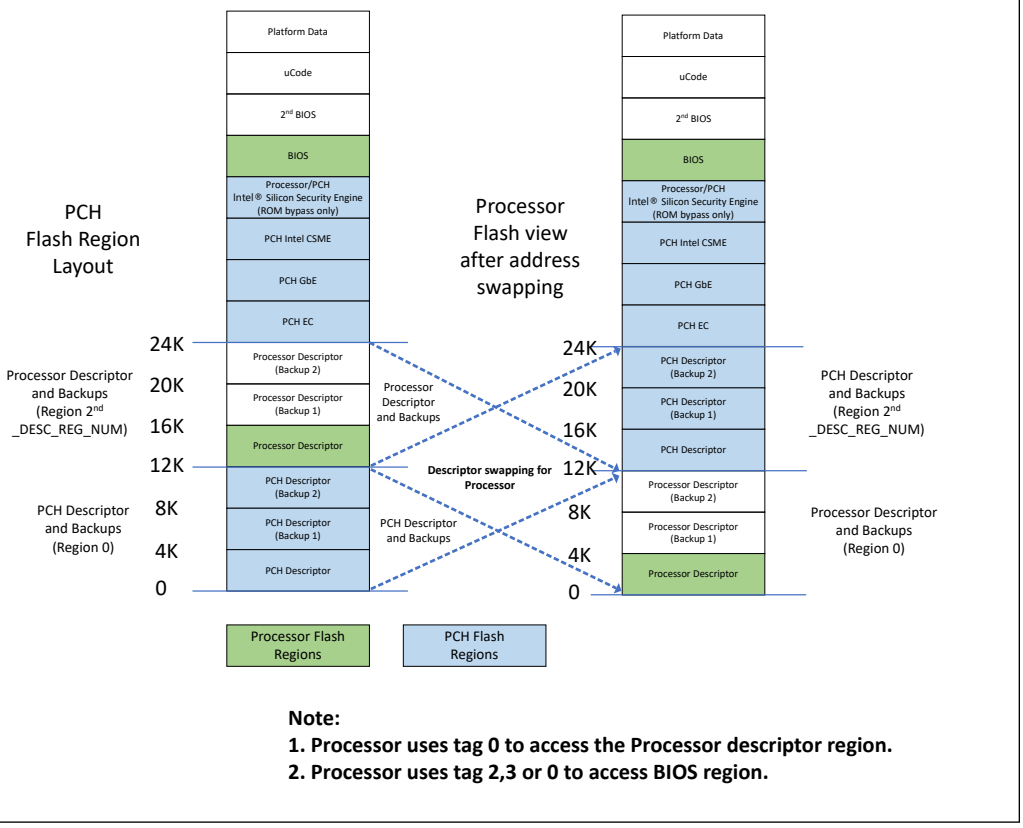
The processor transparently accesses the descriptor. The PCH eSPI controller swaps the flash read/write cycles from the processor with an address within 0-12KB to 12-24KB on PCH flash (MAF or SAF).

From the PCH SPI controller perspective, the PCH descriptor (including main and two backups) is in region 0, located at the flash address 0-12KB, while the processor descriptor (including main and two backups) is in the region located at flash address 12KB-24KB.

From the processor SPI controller perspective, due to the swapping, the processor descriptor (including main and two backups) is in region 0, located at the flash address 0-12KB, while the PCH descriptor (including the main and two backups) is in the region located at flash address 12KB-24KB.

The descriptor regions shown in the diagram below. The two versions of the descriptors use two separate regions to allow independent access permissions control on the processor and the PCH descriptors.

Figure 2. Processor/PCH Descriptor Layout/Swapping

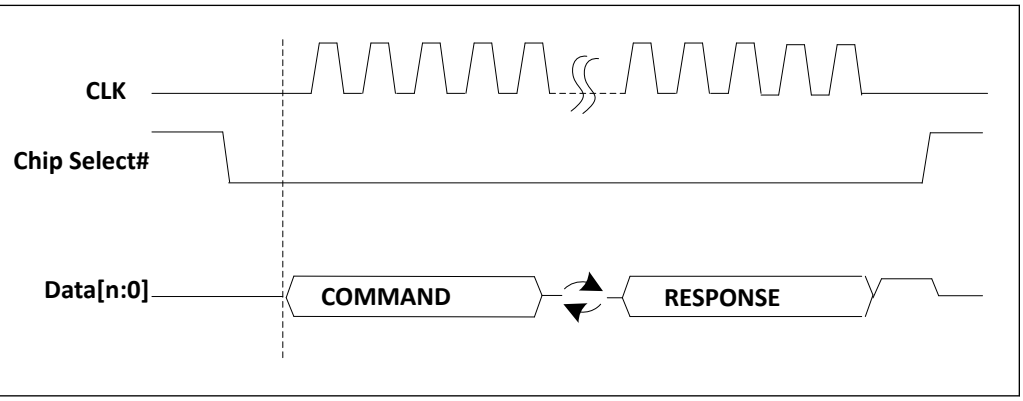


Note:
1. Processor uses tag 0 to access the Processor descriptor region.
2. Processor uses tag 2,3 or 0 to access BIOS region.

8.2 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 25 on page 51).

Figure 3. Basic eSPI Protocol



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

A transaction is initiated by the Processor/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 device.

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 processor while the serial clock is still low and sampled on the rising edge of the clock by the device. Subsequent data is driven on the falling edge of the clock from the processor and sampled on the rising edge of the clock by the device. Data from the device is driven out on the falling edge of the clock and is sampled on a falling edge of the clock by the processor/PCH.

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

8.3 Signal Description

Signal Name	Type	Description
DIR_ESPI_IO0	I/O	**Direct eSPI Signal 0 (PCH)**: Direct eSPI bi-directional command or data between Processor and PCH.
DIR_ESPI_IO1	I/O	**Direct eSPI Signal 1 (PCH)**: Direct eSPI bi-directional command or data between Processor and PCH.
DIR_ESPI_IO2	I/O	**Direct eSPI Signal 2 (PCH)**: Direct eSPI bi-directional command or data between Processor and PCH.
DIR_ESPI_IO3	I/O	**Direct eSPI Signal 3 (PCH)**: Direct eSPI bi-directional command or data between Processor and PCH.
DIR_ESPI_CS#	I	**Direct eSPI Chip Select (PCH):** Driving CS# signal low to enable eSPI PCH for the transaction.
DIR_ESPI_CLK	I	**Direct eSPI Clock (PCH):** eSPI clock output from the Processor to PCH.
DIR_ESPI_RESET#	I	**Direct eSPI Reset (PCH):** Reset signal from the Processor to PCH.
DIR_ESPI_RCLK	O	**Direct eSPI Return Clock (PCH):** Direct eSPI Clock from PCH to Processor.
DIR_ESPI_ALERT#	I	**Direct eSPI Alert Signal (PCH):** Direct eSPI ALERT# driven by PCH to Processor to request service).

8.4 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
DIR_ESPI_IO [3:0]	Pull-up	20 kohm +/- 30%	
DIR_ESPI_CLK	Pull-down	20 kohm+/- 30%	
DIR_ESPI_CS#	Pull-up	20 kohm +/- 30%	

continued...

Signal	Resistor Type	Value	Notes
DIR_ESPI_RCLK	Pull-down	20 kohm +/- 30%	
DIR_ESPI_ALERT#	Pull-up	20 kohm +/- 30%	
DIR_ESPI_RESET#	None	-	

8.5 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5
DIR_ESPI_IO [3:0]	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up
DIR_ESPI_CLK	Primary	Internal Pull-down	Driven Low	Driven Low
DIR_ESPI_RCLK	Primary	Internal Pull-down	Driven Low	Driven Low
DIR_ESPI_ CS#	Primary	Internal Pull-up	Driven High	Driven High
DIR_ESPI_ALERT#	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up
DIR_ESPI_RESET#	Primary	Driven Low	Driven High	Driven High
Note: Reset reference for primary well pins is PCH_RSMRST#.				

9.0 Direct Media Interface (DMI)

The PCH communicates with the processor using high speed DMI that supports 16 GT/s, 8 GT/s, 5 GT/s and 2.5 GT/s data rates.

9.1 Functional Description

The DMI Link supports a maximum of x4 widths. The DMI Supported link width will vary depending on the PCH SKU.

9.1.1 Lane Reversal

x4 and x8 DMI PCH Lane Reversal is supported. OEM can enable this feature in the PCH to ease the DMI link routing (between the processor and the PCH) on the board. PCH DMI Lane Reversal is enabled or disabled through a SoftStrap.

9.1.2 Polarity Inversion

Polarity Inversion is supported on all the Receiver Lanes in PCH DMI. PCH DMI autonomously detects the polarity inversion (Rx+ and Rx- is connected reversed) based on the Training Sequence received and enabled it during Link Training.

9.2 Signal Description

Signal Name	Type	Description
DMI0_TXP DMI0_TXN	O	DMI transmit lane 0
DMI0_RXP DMI0_RXN	I	DMI receive lane 0
DMI1_TXP DMI1_TXN	O	DMI transmit lane 1
DMI1_RXP DMI1_RXN	I	DMI receive lane 1
DMI2_TXP DMI2_TXN	O	DMI transmit lane 2
DMI2_RXP DMI2_RXN	I	DMI receive lane 2
DMI3_TXP DMI3_TXN	O	DMI transmit lane 3
DMI3_RXP DMI3_RXN	I	DMI receive lane 3
DMI4_TXP DMI4_TXN	O	DMI transmit lane 4

continued...

Signal Name	Type	Description
DMI4_RXP DMI4_RXN	I	DMI receive lane 4
DMI5_TXP DMI5_TXN	O	DMI transmit lane 5
DMI5_RXP DMI5_RXN	I	DMI receive lane 5
DMI6_TXP DMI6_TXN	O	DMI transmit lane 6
DMI6_RXP DMI6_RXN	I	DMI receive lane 6
DMI7_TXP DMI7_TXN	O	DMI transmit lane 7
DMI7_RXP DMI7_RXN	I	DMI receive lane 7
DMI_PERST#	I	DMI PERST Strobe
DMI_RCOMP	I	DMI Impedance Compensation Inputs

9.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5	Deep Sx
DMI[7:0]_RXN DMI[7:0]_RXP	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
DMI[7:0]_TXN DMI[7:0]_TXP	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
Note: 1. Reset reference for primary well pins is PCH_RSMRST#.					

9.4 DMI Configuration

Below are the supported DMI configuration options:

Table 26. PCH DMI Link Mapping

PCH DMI Link Width Details			PCH DMI Lane Reversal	PCH DMI Physical Lanes							
SKU	Soft Strap	Negotiated Link Width		0	1	2	3	4	5	6	7
				PCH DMI Logical Lanes							
x4	x4 or x8	x4	No	0	1	2	3				
			Yes	3	2	1	0				
x8	x4	x4	No	0	1	2	3				
			Yes					3	2	1	0
x8	x8	x8	No	0	1	2	3	4	5	6	7
			Yes	7	6	5	4	3	2	1	0

9.5 DMI CLKREQ, DMI REFCLK, DMI PERST#

The PCH has integrated hardware based CLKREQ/REFCLK mechanism to support DMI low power states. This circuitry is aimed to remove the timing/complexity involves in massage based mechanism used on earlier processors. DMI_PERST is also part of this circuitry. For more information, refer to System Clocks on page 153

10.0 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 Host Attached Flash (MAF) and Device Attached Flash (SAF).
- Support for up to 50 MHz (configured by soft straps)
- Up to quad mode support
- Support for PECI over eSPI
- Support for Multiple OOB Controller (dedicated OOB channel for different OOB Controllers in the PCH such as PMC and CSME)
- Transmitting RTC time/date to the device upon request
- In-band messages for communication between the PCH and device to eliminate side-band signals.
- Real time SPI flash sharing, allowing real time operational access by the PCH and device.

Table 27. Acronyms

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

Table 28. References

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

10.1 Functional Description

10.1.1 Operating Frequency

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

10.1.2 WAIT States from eSPI device

There are situations when the device cannot predict the length of the command packet from the controller. For non-posted transactions, the device 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 device after the TAR cycles.

10.1.3 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 device 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 device.

10.1.4 Flash Sharing Mode

eSPI supports both Host and Device Attached Flash sharing (abbreviated in this as MAFS and SAFS, respectively).

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

10.1.5 PECI Over eSPI

When PECI Over eSPI is enabled, the eSPI device (i.e. EC) can access the PCH 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 PCH from Cx and respond back once the data is available from PCH.

10.1.6 Multiple OOB processes

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

10.1.7 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 device is discovered through the GET_CONFIGURATION command issued by the PCH to the eSPI device during initialization.

Table below summarizes the eSPI channels and supported transactions.

Table 29. 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**: 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 device (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 device 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 device. These signals are tunneled between the 2 components over eSPI.
- **Serial IRQ Interrupts**: Interrupts are tunneled from the eSPI device to the PCH. Both edge and triggered interrupts are supported.
- **eSPI Virtual Wires (VW)**

  Table below summarizes the virtual wires in eSPI mode.

Table 30. 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
PRIM_PWRDN_ACK	Output	ESPI_RESET#	No

continued...

Virtual Wire	PCH Pin Direction	Reset Control	Pin Retained in PCH (For Use by Other Components)
SUSWARN#	Output	ESPI_RESET#	No
SUS_ACK	Input	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_S3#/SLP_S4#/SLP_S5#/ SLP_LAN#/SLP_WLAN#	Output	PCH_RSMRST#	Yes
DEVICE_BOOT_LOAD_DONE	Input	ESPI_RESET#	N/A
DEVICE_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
DNX_WARN	Output	PLTRST#	N/A
DNX_ACK	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 eSPI controller will issue a message to the interrupt controller when it receives an IRQ group in its VW packet, indicating a state change for that IRQ line number.

  The eSPI device 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 device. 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 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 device which was used to communicate the MCTP messages. The eSPI controller simply acts as a message transport and forwards the packets between the Intel® CSME and eSPI device.
- **Tunnel PCH Temperature Data to the eSPI device**: The eSPI controller stores the PCH temperature data internally and sends it to the device 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 device**: the eSPI controller captures this data internally at periodic intervals from the PCH RTC controller and sends it to the 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 device. 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 device and the PCH response back are shown in the two figures below.

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

Figure 5. PCH Response to eSPI device 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 Device Addr. = 0Eh (eSPI Device 0/EC)							0
4	Common code = 01h (Get_Temp)							
5	Byte Count = 02h							
6	Source Device Address [7:0] = 01h (OOB HW Handler)							1
7	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 device. This allows the eSPI device 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 device and the PCH response back to the device are shown in the two figures below.

Figure 6. eSPI Device Request to PCH for PCH RTC Time

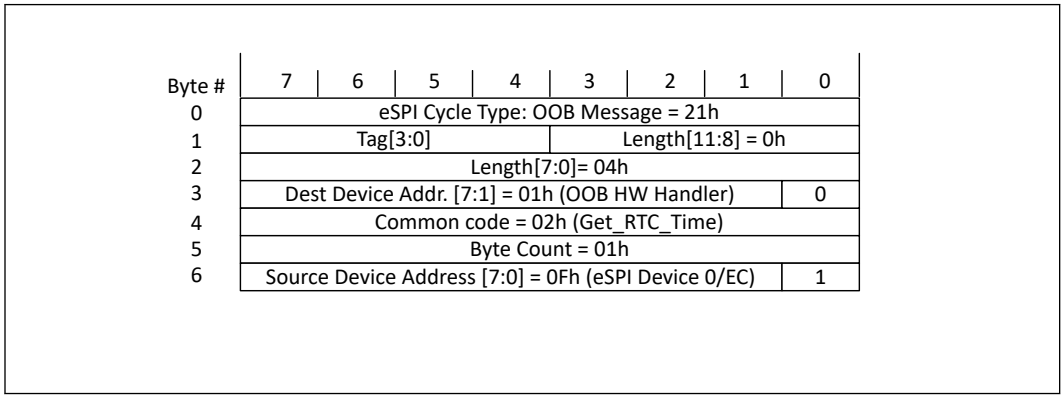


Figure 7. PCH Response to eSPI device with RTC Time

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

NOTES

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

Flash Access Channel (Channel 3) Overview

The Flash Access channel supports the Host 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 Host Attached Flash Channel controller (MAFCC) tunnels flash accesses from eSPI device to the 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 device in a separate completion packet.

- **Host Attached Flash Channel Controller (MAFCC) Flash Operations and Addressing**

  The EC is allocated a dedicated region within the eSPI Host-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.

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

  The PCH is allocated dedicated regions (for each of the supported Controllers) within the eSPI SAFCC. 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.

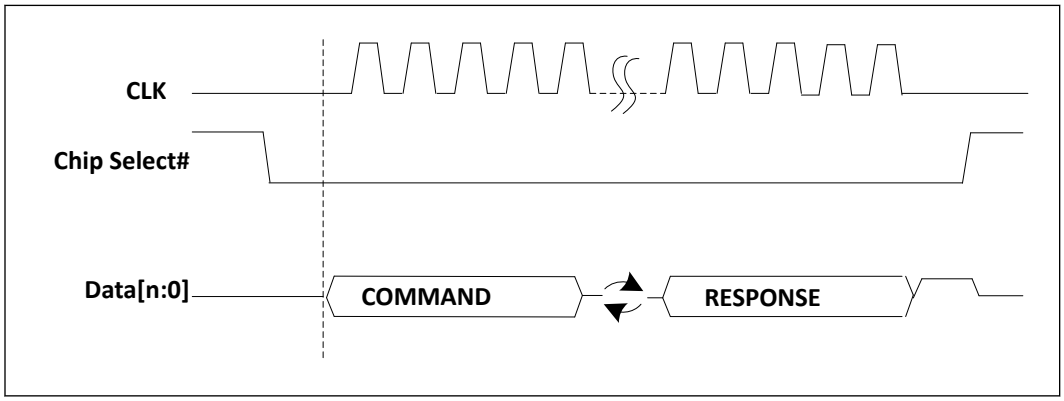
  The Device will optionally perform additional checking on the PCH provided address. In case of an error due to incorrect address or any other issues it will synthesize an unsuccessful completion back to the eSPI Host.

  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.

10.2 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 28 on page 62).

Figure 8. Basic eSPI Protocol



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

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 device.

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 device. 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 device. Data from the device 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).

10.3 Signal Description

Signal Name	Type	Description
GPP_A00/**ESPI_IO0**	I/O	**eSPI Data Signal 0:** Bi-directional pin used to transfer data between the PCH and eSPI device.
GPP_A01/**ESPI_IO1**	I/O	**eSPI Data Signal 1:** Bi-directional pin used to transfer data between the PCH and eSPI device
GPP_A02/**ESPI_IO2**/PRIMWARN#/ PRIMPWRDNACK	I/O	**eSPI Data Signal 2:** Bi-directional pin used to transfer data between the PCH and eSPI device
GPP_A03/**ESPI_IO3**/PRIMACK#	I/O	**eSPI Data Signal 3:** Bi-directional pin used to transfer data between the PCH and eSPI device
GPP_A04/**ESPI_CS0#**	O	**eSPI Chip Select 0:** Driving CS# signal low to select eSPI device for the transaction.
GPP_A05/**ESPI_CLK**	O	**eSPI Clock:** eSPI clock output from the PCH to device.
GPP_A06/**ESPI_RESET#**	O	**eSPI Reset:** Reset signal from the PCH to eSPI device.
GPP_A07/**ESPI_CS1#**	O	**eSPI Chip Select 1 :**Driving CS# signal low to select eSPI device for the transaction.
GPP_A08/**ESPI_CS2#**	O	**eSPI Chip Select 2:**Driving CS# signal low to select eSPI device for the transaction.
GPP_A09/**ESPI_CS3#**	O	**eSPI Chip Select 3:**Driving CS# signal low to select eSPI device for the transaction.
GPP_A10/**ESPI_ALERT0#**	I	**eSPI Alert 0 :**Alert signal from eSPI device to the PCH. *Note:* If only a single device is connected, the eSPI Compatibility Specification requires that the device must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.
GPP_A11/**ESPI_ALERT1#**	I	**eSPI Alert 1 :** Alert signal from eSPI device to the PCH. *Note:* If only a single device is connected, the eSPI Compatibility Spec requires that the device must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.
GPP_A12/**ESPI_ALERT2#**	I	**eSPI Alert 2:** Alert signal from eSPI device to the PCH. *Note:* If only a single device is connected, the eSPI Compatibility Spec requires that the device must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.
GPP_A13/**ESPI_ALERT3#**	I	**eSPI Alert 3:** Alert signal from eSPI device to the PCH.

continued...

Signal Name	Type	Description
		Note: If only a single device is connected, the eSPI Compatibility Spec requires that the device must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.

10.4 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
ESPI_IO[3:0]	Pull-up	20 kohm +/- 30%	
ESPI_CLK	Pull-down	20 kohm+/- 30%	
ESPI_ CS0#	Pull-up	20 kohm +/- 30%	

10.5 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5
ESPI_IO[3:0]	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up
ESPI_CLK	Primary	Internal Pull-down	Driven Low	Driven Low
ESPI_ CS0#	Primary	Internal Pull-up	Driven High	Driven High
ESPI_RESET#	Primary	Driven Low	Driven High	Driven High
Note: Reset reference for primary well pins is PCH_RSMRST#.				

11.0 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 31. Acronyms

Acronyms	Description
GbE	Gigabit Ethernet

Table 32. References

Specification	Location
IEEE 802.3 Ethernet	http://standards.ieee.org/getieee802/
Intel® Ethernet Connection I219 Datasheet	http://www.intel.com/content/www/us/en/

11.1 Signal Description

Table 33. GbE LAN Signals

Signal Name	Type	Description
PCIE_7_TXP/**GbE_0B_TXP** PCIE_7_TXN/**GbE_0B_TXN** PCIE_3_TXP/**GbE_0A_TXP** PCIE_3_TXN/**GbE_0A_TXN** PCIE_15_TXP/SATA_2_TXP/**GbE_0C_TXP** PCIE_15_TXN/SATA_2_TXN/**GbE_0C_TXN**	O	Differential transmit pairs to the Intel® Ethernet Connection I219 based on the PCIe interface. Refer to PCI Express* (PCIe*) on page 120 for details on the PCI Express*transmit signals.
PCIE_7_RXP/**GbE_0B_RXP** PCIE_7_RXN/**GbE_0B_RXN** PCIE_3_RXP/**GbE_0A_RXP** PCIE_3_RXN/**GbE_0A_RXN** PCIE_15_RXP/SATA_2_RXP/**GbE_0C_RXP** PCIE_15_RXN/SATA_2_RXN/ **GbE_0C_RXN**	I	Differential receive pairs to the Intel® Ethernet Connection I219 based on the PCIe interface. Refer to PCI Express* (PCIe*) on page 120 for details on the PCI Express* transmit signals.
GPP_D10/**SML0DATA**	I/OD	System Management Link data signal interface to Intel® Ethernet Connection I219. Refer to System Management Interface and SMLink on page 156 for details on the SML0DATA signal. *Note:* The Intel® Ethernet Connection I219 connects to SML0DATA signal.
GPP_D09/**SML0CLK**	I/OD	System Management Link data signal interface to Intel® Ethernet Connection I219. Refer to System Management Interface and SMLink on page 156 for details on the SML0CLK signal. *Note:* The Intel® Ethernet Connection I219 connects to SML0CLK signal.

continued...

Signal Name	Type	Description
GPD11/**LANPHYPC**	O	**LAN PHY Power Control:** LANPHYPC should be connected to LAN_DISABLE_N on the PHY. The 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.
GPD09/**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.

11.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
LAN_WAKE#	External Pull-up required.	4.7 kohm +/- 5%	10 kohm +/- 5% pull-up resistor is also acceptable.

11.3 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. Some 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.

NOTE

Refer the section "Flexible High Speed I/O" for GbE lane allocation options.

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 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 host 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

11.3.1 GbE PCI Bus Interface

The GbE controller has a PCI 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 64 bytes or less and are aligned to make better use of memory controller resources.

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.

11.3.2 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.

11.3.3 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 PCHand 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 34. 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	SMLink0 / Wake#	
Manageability	S0ix / Sx	SMLink0	

11.3.4 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 S3 and S4 to S0.

The integrated GbE controller contains power management registers for PCI and supports D0 and D3 states. PCI 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 203.

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 , 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 not required if the source of power is battery, the host BIOS must set DC_PP_DIS.

12.0 Host System Management Bus (SMBus) Controller

The PCH provides a System Management Bus (SMBus) 2.0 host controller as well as an SMBus Device 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 35. Acronyms

Acronyms	Description
ARP	Address Resolution Protocol
CRC	Cyclic Redundancy Check
SMBus	System Management Bus

Table 36. References

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

12.1 Functional Description

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

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

12.1.1 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 Target 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 Target 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.

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

Using the SMB host controller to send commands to the PCH SMB Target 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 Target 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 Target Address and Device Command Registers are sent. For the Receive Byte command, the Transmit Target 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 Target 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 Target device. Then it must follow that command with a repeated start condition to denote a read from that device's address. The target 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 Target 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 target 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 Target 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 Target 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 target 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 target 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 target 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 target 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 Target 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 37. I²C* Block Read

Bit	Description
1	Start
8:2	Target Address – 7 bits
9	Write
10	Acknowledge from target
18:11	Send DATA1 register
19	Acknowledge from target
20	Repeated Start
27:21	Target Address – 7 bits
28	Read
29	Acknowledge from target
37:30	Data byte 1 from target – 8 bits
38	Acknowledge
46:39	Data byte 2 from target – 8 bits
47	Acknowledge
–	Data bytes from target / Acknowledge
–	Data byte N from target – 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 target 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 controller 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 target 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 target 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 controllers 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 controller 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 controller 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 controller if it is not ready to send or receive data.

Bus Timeout (PCH as SMBus controller)

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 target 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 38. 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 (Target 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	Target SMI# generated (SMBUS_SMI_STS)
	1	0	0	Interrupt generated

Table 39. Enables for SMBus Target 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
Target Write to Wake/ SMI# Command	X	X	Wake generated when asleep. Target SMI# generated when awake (SMBUS_SMI_STS).
Target Write to SMLINK_SLAVE_SMI Command	X	X	Target 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 40. Enables for the Host Notify Command

HOST_NOTIFY_INTREN (Target Control I/O Register, Offset 11h, Bit 0)	SMB_SMI_EN (Host Configuration Register, D31:F4:Off40h, Bit 1)	HOST_NOTIFY_WKEN (Target 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	Target 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.

12.1.2 SMBus Target Interface

The PCH SMBus Target interface is accessed using the SMBus. The SMBus target logic will not generate or handle receiving the PEC byte and will only act as a Legacy Alerting Protocol device. The target 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 Target Address register: This is the address that the PCH decodes. A default value is provided so that the target interface can be used without the processor having to program this register.
- Receive Target 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 target logic caused an interrupt or SMI# Bit 0 of the target 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 target logic until either:

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

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

Format of Target Write Cycle

The external controller performs Byte Write commands to the PCH SMBus Target 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 41. Target 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 42. 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 controller performs Byte Read commands to the PCH SMBus Target 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 43. Target Read Cycle Format

Bit	Description	Driven By	Comment
1	Start	External Micro controller	
2–8	Target Address - 7 bits	External Micro controller	Must match value in Receive Target 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	Target Address - 7 bits	External Micro controller	Must match value in Receive Target 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 44. Data Values for Target 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	Reserved
	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 PLT_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 target 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 Target 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 Target Address, it will assume that the protocol is followed, ignore bit 28, and proceed with the Target Read cycle.

Target Read of RTC Time Bytes

The PCH SMBus target interface allows external SMBus controller 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 target 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 Target interface only supports Byte Read operation. The external SMBus controller 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 controller 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 controller 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 45. Host Notify Format

Bit	Description	Driven By	Comment
1	Start	External Controller	
8:2	SMB Host Address – 7 bits	External Controller	Always 0001_000
9	Write	External Controller	Always 0
10	ACK (or NACK)	PCH	PCH NACKs if HOST_NOTIFY_STS is 1
17:11	Device Address – 7 bits	External Controller	Indicates the address of the controller ; loaded into the Notify Device Address Register
18	Unused – Always 0	External Controller	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 Controller	Loaded into the Notify Data Low Byte Register
28	ACK	PCH	
36:29	Data Byte High – 8 bits	External Controller	Loaded into the Notify Data High Byte Register
37	ACK	PCH	
38	Stop	External Controller	

Format of Read Command

The external controller performs Byte Read commands to the PCH SMBus Target 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 46. Target Read Cycle Format

Bit	Description	Driven By	Comment
1	Start	External Micro controller	
2–8	Target Address - 7 bits	External Micro controller	Must match value in Receive Target 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 Tale below for a list of implemented registers.
19	ACK	PCH	
20	Repeated Start	External Micro controller	
21–27	Target Address - 7 bits	External Micro controller	Must match value in Receive Target 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 47. Data Values for Target 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

continued...

Register	Bits	Description
	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 PLT_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

continued...

Register	Bits	Description
E	7:0	Month of the RTC
F	7:0	Year of the RTC
10h–FFh	7:0	Reserved

Table 48. Enables for SMBus Target 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
Target Write to Wake/SMI# Command	X	X	Wake generated when asleep. Target SMI# generated when awake (SMBUS_SMI_STS)
Target Write to SMLINK_SLAVE_SMI Command	X	X	Target 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

12.2 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.

12.3 Signal Description

Signal Name	Type	Description
GPP_C00/**SMBCLK**/USB-C_GPP_C00	I/OD	**SMBus Clock:** External Pull-up resistor is required.
GPP_C01/**SMBDATA**/USB-C_GPP_C01	I/OD	**SMBus Data:** External Pull-up resistor is required.
GPP_C02/**SMBALERT#**/ USB-C_GPP_C02	I/OD	**SMBus Alert:** This signal is used to wake the system or generate SMI#. External Pull-up resistor is required.

12.4 Integrated Pull-Ups and Pull-Downs

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

12.5 I/O Signal Planes and States

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

13.0 Integrated Sensor Hub (ISH)

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, I^3C, 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.

Table 49. 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

Table 50. References

Specification	Location
I²C Specification Version 6.0	http://www.nxp.com/docs/en/user-guide/UM10204.pdf

13.1 Features

13.1.1 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:

- Host I²C operation
- Support for the following operating speeds:
  - Standard mode: 100 kbps
  - Fast Mode: 400 kbps
  - Fast Mode Plus: 1000 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

13.1.2 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:

- Operate in Host 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)

13.1.3 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.

13.1.4 ISH GPIOs

The ISH supports eight dedicated GPIOs.

13.2 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).

13.2.1 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.

13.2.2 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 640 KB 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.

13.2.3 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 Con figuration space. It is used only in ACPI mode (that is, when the PCI con figuration 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 Processor.

PCI Power Management

PME is not supported in ISH.

13.2.4 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 51. 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.

13.2.5 ISH Interrupt Handling via IOAPIC (Interrupt Controller)

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

13.3 Signal Description

Signal Name	Type	Description
GPP_I14/**ISH_I2C0_SDA**/**ISH_I3C0_SDA**	I/OD	ISH I^2C 0 Data ISH I^3C 0 Data
GPP_I15/**ISH_I2C0_SCL**/**ISH_I3C0_SCL**	I/OD	ISH I^2C 0 Clk ISH I^3C 0 Clk
GPP_H18/**ISH_I2C1_SDA**/SMI#/USB-C_GPP_H18	I/OD	ISH I^2C 1 Data
GPP_H19/**ISH_I2C1_SCL**/NMI#/USB-C_GPP_H19	I/OD	ISH I^2C 1 Clk
GPP_C06/**ISH_I2C2_SDA**/I2C3_SDA/USB-C_GPP_C06	I/OD	ISH I^2C 2 Data
GPP_C07/**ISH_I2C2_SCL**/I2C3_SCL/USB-C_GPP_C07	I/OD	ISH I^2C 2 Clk
GPP_B08/**ISH_GP0**	I/O	ISH GPIO 0
GPP_B09/**ISH_GP1**	I/O	ISH GPIO 1

continued...

Signal Name	Type	Description
GPP_B10/**ISH_GP2**	I/O	ISH GPIO 2
GPP_B15/**ISH_GP3**	I/O	ISH GPIO 3
GPP_B16/**ISH_GP4**	I/O	ISH GPIO 4
GPP_B17/**ISH_GP5**	I/O	ISH GPIO 5
GPP_B05/SX_EXIT_HOLDOFF#/**ISH_GP6**	I/O	ISH GPIO 6
GPP_B07/**ISH_GP7**	I/O	ISH GPIO 7
GPP_B02/**ISH_GP8**	I/O	ISH GPIO 8
GPP_E11/**ISH_GP9**	I/O	ISH GPIO 9
GPP_E12/**ISH_GP10**	I/O	ISH GPIO 10
GPP_I16_RSVD/**ISH_GP11**	I/O	ISH GPIO 11
GPP_C04/**ISH_UART0_TXD**/SML0BCLK/I2C2_SCL/USB-C_GPP_C04	O	ISH UART 0 Transmit Data
GPP_C03/**ISH_UART0_RXD**/SML0BDATA/I2C2_SDA/USB-C_GPP_C03	I	ISH UART 0 Receive Data
GPP_B01/**ISH_UART0_RTS#**/GSPI2_CS1#	O	ISH UART 0 Request To Send
GPP_E21/**ISH_UART0_CTS#**/SML0BALERT#	I	ISH UART 0 Clear to Send
GPP_C13/UART1_TXD/**ISH_UART1_TXD**/USB-C_GPP_C13	O	ISH UART 1 Transmit Data
GPP_C12/UART1_RXD/**ISH_UART1_RXD**/USB-C_GPP_C12	I	ISH UART 1 Receive Data
GPP_C14/UART1_RTS#/**ISH_UART1_RTS#**/USB-C_GPP_C14	O	ISH UART 0 Request To Send
GPP_C15/UART1_CTS#/**ISH_UART1_CTS#**/USB-C_GPP_C15	I	ISH UART 1 Clear to Send
GPP_R09/**ISH_SPI_CS#**/GSPI2_CS0#	O	ISH SPI Chip Select
GPP_R10/**ISH_SPI_CLK**/GSPI2_CLK	O	ISH SPI Clock
GPP_R11/**ISH_SPI_MISO**/GSPI2_MISO	I	ISH SPI MISO
GPP_R12/**ISH_SPI_MOSI**/GSPI2_MOSI	O	ISH SPI MOSI

13.4 Integrated Pull-Ups and Pull-Down

NA

13.5 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5
ISH_I2C0_SDA	Primary	Undriven	Undriven	Undriven
ISH_I2C0_SCL	Primary	Undriven	Undriven	Undriven
ISH_I2C1_SDA	Primary	Undriven	Undriven	Undriven
ISH_I2C1_SCL	Primary	Undriven	Undriven	Undriven

continued...

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5
ISH_I2C2_SDA	Primary	Undriven	Undriven	Undriven
ISH_I2C2_SCL	Primary	Undriven	Undriven	Undriven
ISH_GP[11:0]	Primary	Undriven	Undriven	Undriven
ISH_UART0_TXD	Primary	Undriven	Undriven	Undriven
ISH_UART0_RXD	Primary	Undriven	Undriven	Undriven
ISH_UART0_RTS#	Primary	Undriven	Undriven	Undriven
ISH_UART0_CTS#	Primary	Undriven	Undriven	Undriven
ISH_UART1_TXD	Primary	Undriven	Undriven	Undriven
ISH_UART1_RXD	Primary	Undriven	Undriven	Undriven
ISH_UART1_CTS#	Primary	Undriven	Undriven	Undriven
ISH_UART1_RTS#	Primary	Undriven	Undriven	Undriven
ISH_SPI_CS#	Primary	Undriven	Undriven	Undriven
ISH_SPI_CLK	Primary	Undriven	Undriven	Undriven
ISH_SPI_MISO	Primary	Undriven	Undriven	Undriven
ISH_SPI_MOSI	Primary	Undriven	Undriven	Undriven
Note: 1. Reset reference for primary well pins is PCH_RSMRST#.				

14.0 Intel® Serial I/O Improved Inter-Integrated Circuit (I3C) Controllers

I3C specification is backward compatible with I2C devices. The I3C enables dynamic address allocation and inband interrupts. The Spec also allows for hot-plug / hot-join of devices. The I3C Specification is backward compatible with legacy I2C devices and enables coexistence of legacy I2C and I3C devices on the same bus in Fast Mode, Fast Mode Plus modes, without clock stretching. The PCH has one I3C controller compliant to MIPI I3C HCI Specification, that can support 2 I3C buses and up to 8 devices per bus (subject to meeting electrical/topology requirements).

The I3C interfaces support the following features:

- Support for MIPI I3C spec v1.0, and MIPI I3C HCI Specification
- Speed: standard mode (up to 100 Kb/s), fast mode (up to 400 Kb/s), fast mode plus (up to 1 MB/s)
- Supports clock loopback using dummy IO to meet ACIO timing
- Maximum theoretical Baud rate is 12900 kbps
- Maximum validated Baud rate is 12500 kbps
- Operate in 1.8 V Only
- DMA support with 64-byte DMA FIFO per channel (up to 32-byte burst)
- 64-byte Tx FIFO and 64-byte Rx FIFO
- PME/wake support for IBI, when in S0ix
- PCI/ACPI enumeration support
- I3C static addressing and dynamic addressing support
- I3C in-band interrupt
- I3C transactions using SDR
- Error detection and recovery methods M0, M2
- For stalling Host clock on data buffering
- Host I3C operation only

NOTES

1. The controllers must only be programmed to operate in Host mode only. I3C Device mode is not supported.
2. I3C multi Hosts is not supported.
3. Simultaneous configuration of Fast Mode and Fast Mode Plus is not supported.

Table 52. Acronyms

Acronyms	Description
I3C	Improved Inter-Integrated Circuit
SCL	Serial Clock Line
SDA	Serial Data Line

14.1 Functional Description

14.1.1 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 I3C peripheral deadlock condition where the reset goes active in the middle of a transaction, the I3C controller must be idle before a reset can be initiated.

14.1.2 Power Management

Device Power Down Support

To power down peripherals connected to I3C 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 I3C bus is powered off (power gated). The 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.

14.1.3 Interrupts

I3C 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.

14.2 Signal Description

Signal Name	Type	Description
GPP_I02/I2C0_SDA/**I3C0_SDA**	I/OD	**I3C Link 0 Serial Data Line**
GPP_I03/I2C0_SCL/**I3C0_SCL**	I/OD	**I3C Link 0 Serial Clock Line**

14.3 Integrated Pull-Ups and Pull-Downs

None.

14.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5
I3C0_SDA	Primary	Undriven	Undriven	Undriven
I3C0_SCL	Primary	Undriven	Undriven	Undriven
Note: 1. Reset reference for primary well pins is PCH_RSMRST#.				

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

The implements six I²C controllers for six independent I²C interfaces, I2C0-I2C5. Each interface is a two-wire serial interface consisting of a serial data line (SDA) and a serial clock (SCL).

The I²C 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).
- Operate in 1.8 V or 3.3 V
- Host I²C operation only
- 7-bit or 10-bit addressing
- 7-bit or 10-bit combined format transfers
- Bulk transmit mode
- Ignoring CBUS addresses (an older ancestor of I²C used to share the I²C bus)
- Interrupt or polled-mode operation
- Bit and byte waiting at all bus speed
- Component parameters for configurable software driver support
- Programmable SDA hold time (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 Host mode only. I²C device mode is not supported.
2. I²C multi hosts is not supported.
3. Simultaneous configuration of Fast Mode and Fast Mode Plus/High speed mode is not supported.
4. I²C General Call is not supported.

Table 53. Acronyms

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

Table 54. References

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

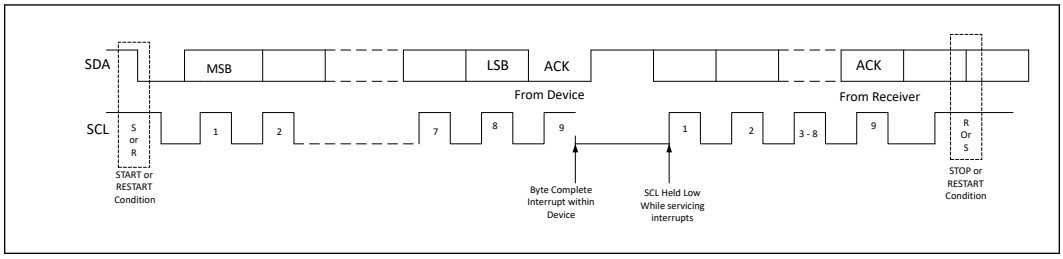
15.1 Functional Description

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

Figure 9. Data Transfer on I²C Bus



Combined Formats

The I²C controllers support mixed read and write combined format transactions in both 7-bit and 10-bit addressing modes.

The 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 host, when the controller completes an I²C 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 I²C transfer completes, a STOP is issued and the next transfer is issued following a START condition.

15.1.2 DMA Controller

The I²C controllers 0 to 3 (I2C0 - I2C3) each has 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 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 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.

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

15.1.4 Power Management

Device Power Down Support

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

15.1.5 Interrupts

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

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

15.1.6 Error Handling

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

15.1.7 Programmable SDA Hold Time

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

15.2 Signal Description

Signal Name	Type	Description
GPP_I02/**I2C0_SDA**/I3C0_SDA	I/OD	**I^2C Link 0 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_I03/**I2C0_SCL**/I3C0_SCL	I/OD	**I^2C Link 0 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_I04/**I2C1_SDA**	I/OD	**I^2C Link 1 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_I05/**I2C1_SCL**	I/OD	**I^2C Link 1 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_C03/ISH_UART0_RXD/ SML0BDATA/**I2C2_SDA**/USB-C_GPP_C03	I/OD	**I^2C Link 2 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_C04/ISH_UART0_TXD/ SML0BCLK/**I2C2_SCL**/USB-C_GPP_C04	I/OD	**I^2C Link 2 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_C06/ISH_I2C2_SDA/**I2C3_SDA**/ USB-C_GPP_C06	I/OD	**I^2C Link 3 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_C07/ISH_I2C2_SCL/**I2C3_SCL**/ USB-C_GPP_C07	I/OD	**I^2C Link 3 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_C16/USB_OC4#/**I2C4_SDA**/ USB-C_GPP_C16	I/OD	**I^2C Link 4 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_C17/USB_OC5#/**I2C4_SCL**/USB-C_GPP_C17	I/OD	**I^2C Link 4 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_C18/USB_OC6#/**I2C5_SDA**/ USB-C_GPP_C18	I/OD	**I^2C Link 5 Serial Data Line** External Pull-up resistor may be required depending on Bus Capacitance.
GPP_C19/USB_OC7#/**I2C5_SCL**/USB-C_GPP_C19	I/OD	**I^2C Link 5 Serial Clock Line** External Pull-up resistor may be required depending on Bus Capacitance.

15.3 Integrated Pull-Ups and Pull-Downs

None.

15.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5
I2C[5:0]_SDA	Primary	Undriven	Undriven	Undriven
I2C[5:0]_SCL	Primary	Undriven	Undriven	Undriven
Note: 1. Reset reference for primary well pins is PCH_RSMRST#.				

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

The 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 55. Acronyms

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

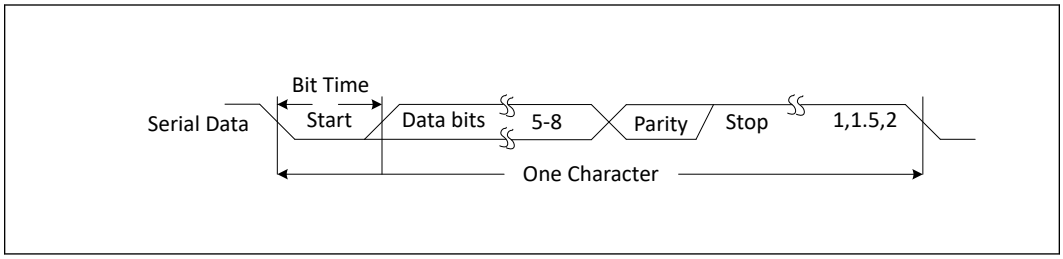
16.1 Functional Description

16.1.1 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 10. UART Serial Protocol



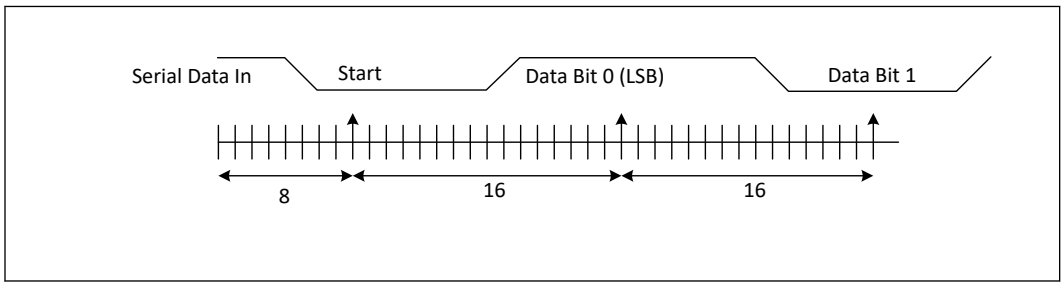
The UART Host Controller Line Control Register (LCR) is used to control the serial character characteristics. The individual bits of the data word are sent after the Start bit, starting with the least significant bit (LSB). These are followed by the optional parity bit, followed by the Stop bit(s), which 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 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 11. UART Receiver Serial Data Sample Points



16.1.2 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.

16.1.3 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 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.

16.1.4 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.

16.1.5 Power Management

Device Power Down Support

In order to power down peripherals connected to 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 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.

16.1.6 Interrupts

UART interface has the ability to interrupt and 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.

16.1.7 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.

16.2 Signal Description

Signal Name	Type	Description
GPP_C08/**UART0_RXD**/ USB-C_GPP_C08	I	**UART 0 Receive Data**
GPP_C09/**UART0_TXD**/ USB-C_GPP_C09	O	**UART 0 Transmit Data**
GPP_C10/**UART0_RTS#**/ USB-C_GPP_C10	O	**UART 0 Request to Send**
GPP_C11/**UART0_CTS#**/ USB-C_GPP_C11	I	**UART 0 Clear to Send**
GPP_C12/**UART1_RXD**/ ISH_UART1_RXD/USB-C_GPP_C12	I	**UART 1 Receive Data**
GPP_C13/**UART1_TXD**/ ISH_UART1_TXD/USB-C_GPP_C13	O	**UART 1 Transmit Data**
GPP_C14/**UART1_RTS#**/ ISH_UART1_RTS#/USB-C_GPP_C14	O	**UART 1 Request to Send**
GPP_C15/**UART1_CTS#**/ ISH_UART1_CTS#/USB-C_GPP_C15	I	**UART 1 Clear to Send**
GPP_C20/**UART2_RXD**/ USB-C_GPP_C20	I	**UART 2 Receive Data**

continued...

Signal Name	Type	Description
GPP_C21/**UART2_TXD**/ USB-C_GPP_C21	O	**UART 2 Transmit Data**
GPP_C22/**UART2_RTS#**/ USB-C_GPP_C22	O	**UART 2 Request to Send**
GPP_C23/**UART2_CTS#**/ USB-C_GPP_C23	I	**UART 2 Clear to Send**
GPP_D20/**UART3_RXD**/ THC1_SPI2_IO2	I	**UART 3 Receive Data**
GPP_D21/**UART3_TXD**/ THC1_SPI2_IO3	O	**UART 3 Transmit Data**
GPP_D22/**UART3_RTS#**/ THC1_SPI2_RST#	O	**UART 3 Request to Send**
GPP_D23/**UART3_CTS#**/ THC1_SPI2_INT#	I	**UART 3 Clear to Send**

16.3 Integrated Pull-Ups and Pull-Downs

None.

16.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5
UART[2:0]_RXD	Primary	Undriven	Undriven	Undriven
UART[2:0]_TXD	Primary	Undriven	Undriven	Undriven
UART2_RTS# **UART0_RTS#**	Primary	Undriven	Undriven	Undriven
UART2_CTS# **UART0_CTS#**	Primary	Undriven	Undriven	Undriven
Note: 1. Reset reference for primary well pins is RSMRST#.				

17.0 Intel® Serial IO Generic SPI (GSPI) Controllers

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

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

The GSPI interfaces support the following features:

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

NOTE

Device mode is not supported.

Table 56. Acronyms

Acronyms	Description
GSPI	Generic Serial Peripheral Interface
LTR	Latency Tolerance Reporting

17.1 Functional Description

17.1.1 Controller Overview

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

The processor or DMA accesses data through the GSPI ports 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 Processor 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 host, 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.

17.1.2 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.

17.1.3 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.

17.1.4 Power Management

Device Power Down Support

In order to power down peripherals connected to the 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 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.

17.1.5 Interrupts

Each interface has the ability to interrupt and 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.

17.1.6 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.

17.2 Signal Description

Signal Name	Type	Description
GPP_I06/**GSPI0_CS0#**	O	**Generic SPI 0 Chip Select 0**
GPP_I00/**GSPI0_CS1#**	O	**Generic SPI 0 Chip Select 1**
GPP_I07/**GSPI0_CLK**	O	**Generic SPI 0 Clock**
GPP_I08/**GSPI0_MISO**	I	**Generic SPI 0 MISO**
GPP_I09/**GSPI0_MOSI**	O	**Generic SPI 0 MOSI** *Note:* This signal is also utilized as a strap. Refer to the pin strap section for more information.
GPP_I10/**GSPI1_CS0#**	O	**Generic SPI 1 Chip Select 0**
GPP_I01/**GSPI1_CS1#**		**Generic SPI 1 Chip Select 1**
GPP_I11/**GSPI1_CLK**	O	**Generic SPI 1 Clock**
GPP_I12/**GSPI1_MISO**	I	**Generic SPI 1 MISO**
GPP_I13/**GSPI1_MOSI**	O	**Generic SPI 1 MOSI** *Note:* This signal is also utilized as a strap. Refer to the pin strap section for more information.
GPP_R09/ISH_SPI_CS#/ **GSPI2_CS0#**	O	**Generic SPI 2 Chip Select 0**
GPP_B01/ISH_UART0_RTS#/ **GSPI2_CS1#**	O	**Generic SPI 2 Chip Select 1**
GPP_R10/ISH_SPI_CLK/ **GSPI2_CLK**	O	**Generic SPI 2 Clock**
GPP_R11/ISH_SPI_MISO/ **GSPI2_MISO**	I	**Generic SPI 2 MISO**
GPP_R12/ISH_SPI_MOSI/ **GSPI2_MOSI**	O	**Generic SPI 2 MOSI** *Note:* This signal is also utilized as a strap. Refer to the pin strap section for more information.
GPP_D16/**GSPI3_CS0#**/ THC1_SPI2_CS#	O	**Generic SPI 3 Chip Select 0**

continued...

Signal Name	Type	Description
GPP_D17/**GSPI3_CLK**/ THC1_SPI2_CLK	O	**Generic SPI 3 Clock**
GPP_D18/**GSPI3_MISO**/ THC1_SPI2_IO0	I	**Generic SPI 3 MISO**
GPP_D19/**GSPI3_MOSI**/ THC1_SPI2_IO1	O	**Generic SPI 3 MOSI** *Note:* This signal is also utilized as a strap. Refer to the pin strap section for more information.

17.3 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
GSPI0_MOSI	Pull Down	20 kohm± 30%	The integrated pull down is disabled after PLT_PWROK assertion
GSPI1_MOSI	Pull Down	20 kohm ± 30%	The integrated pull down is disabled after PLT_PWROK assertion
GSPI2_MOSI	Pull Down	20 kohm ± 30%	The integrated pull down is disabled after PLT_PWROK assertion
GSPI3_MOSI	Pull Down	20 kohm ± 30%	The integrated pull down is disabled after PLT_PWROK assertion
GSPI0_MISO	Pull Down	20 kohm ± 30%	
GSPI1_MISO	Pull Down	20 kohm ± 30%	
GSPI2_MISO	Pull Down	20 kohm ± 30%	
GSPI3_MISO	Pull Down	20 kohm ± 30%	

17.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5
GSPI0_CS0#, GSPI1_CS0# , GSPI2_CS0# GSPI3_CS0#	Primary	Undriven	Undriven	Undriven
GSPI0_CLK, GSPI1_CLK , GSPI2_CLK GSPI3_CLK	Primary	Undriven	Undriven	Undriven
GSPI0_MISO, GSPI1_MISO , GSPI2_MISO GSPI3_MISO	Primary	Undriven	Undriven	Undriven
GSPI0_MOSI, GSPI1_MOSI, GSPI2_MOSI GSPI3_MOSI	Primary	Internal Pull-down	Driven Low	Internal Pull-down
Note: 1. Reset reference for primary well pins is PCH_RSMRST#.				

18.0 PCI Express* (PCIe*)

Table 57. Acronym

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

Table 58. Reference

Specification	Location
PCI Express® Base Specification Revision 5.0, Version 1.0, 22 May 2019	https://pcisig.com/
PCI Express® M.2 Specification Revision 4.0, Version 1.1, April 14 2022	https://pcisig.com/
PCI Express® Card Electromechanical Specification, Revision 5.0, Version 1.0, June 9 2021	https://pcisig.com/

18.1 Functional Description

Table 59. Features Supported

PCIe Controller Feature	PCH-S Controllers
	[6:1]
L1 Sub-States (L1.0, L1.1, L1.2)	Yes
L0s Link State (RX/TX)	No
S3/S4/S5 Sleep States (Sx)	Yes
Common Clock Mode	Yes
Separate Reference Clock with Independent SSC (SRIS)	No
Separate Reference Clock with No SSC (SRNS)	No
Precision Time Management (PTM)	Yes
Advanced Error Reporting (AER)	Yes
End-to-End Lane Reversal	Yes
Latency Tolerance Reporting (LTR)	Yes
PCIe TX Half Swing	No
PCIe TX Full Swing	Yes
Run Time D3 (RTD3)	Yes
Access Control Services (ACS)	Yes
Alternative Routing-ID Interpretation (ARI)	Yes
Port 80h Decode	Yes

continued...

PCIe Controller Feature	PCH-S Controllers [6:1]
Lane Polarity Inversion	Yes
PCIe Controller Root Port Hot-Plug	Yes
Downstream Port Containment (DPC)	No
Enhanced Downstream Port Containment (eDPC)	No
Virtual Channel (VC)	0
NVMe Cycle Router	No
Volume Management Device (Intel® VMD)	Yes
RAID[0, 1, 5, 10] Mode Support[1,2,3]	See PCH SKU details in Introduction section
PCIe Controller (PC) Root Port (RP) Peer-2-Peer (P2P) Mem Write Transactions	RPs between and within PCH PC1/2/3/4/5/6 = No RPs between Processor PC and PCH PC1/2/3/4/5/6 = No
PCIe Controller (PC) Root Port (RP) Peer-2-Peer (P2P) Mem Read Transactions	No
PCIe Controller (PC) Root Port (RP) Peer-2-Peer (P2P) MCTP VDM Transactions	RPs between and within PCH PC1/2/3/4/5/6 = Yes RPs between Processor PC and PCH PC1/2/3/4/5/6 = Yes
PCIe Root Port Initiated Dynamic Width Change	No
PCIe Root Port Initiated Dynamic Speed Change	Yes
End Point Device Initiated Dynamic Width Change	Yes
End Point Device Initiated Dynamic Speed Change	Yes
Flattening Portal Bridge (FPB)	No

NOTES

1. No restrictions on PCIe Controller. PCIe RAID is expected to work across all Root Ports within a PCIe Controller and between Root Ports from different PCIe Controllers (Processor and PCH).
2. No RAID support between PCIe and SATA storage devices.
3. The Intel® Rapid Storage Technology (RST) does not restrict RAID modes so if any unsupported RAID mode is enabled it is up to the motherboard designer to validate any non-supported RAID mode.

18.1.1 PCI Express* Power Management

S4/S5 Sleep State Support

Software initiates the transition to S4/S5 by performing an IO write to the Power Management Controller. After the IO write completion has been returned 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 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.

Latency Tolerance Reporting (LTR)

The PCIe Controller Root Ports support the extended Latency Tolerance Reporting (LTR) capability. LTR provides a means for device endpoints to dynamically report their service latency requirements for memory reads and write access to the Root Ports through the Latency Tolerance Reporting messages. Endpoint devices should transmit a new LTR message to the Root Ports initially during boot and each time its latency tolerance changes. This latency information allows the Power Management Controller (PMC) to make effective and accurate decisions to transition the platform to deeper power management states without the cost of making the wrong decision, since deeper power management states are usually associated with longer exit latency.

18.1.2 Port 80h 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 read or write is 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 80h cycles.

18.1.3 Advanced Error Reporting

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

18.1.4 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.

18.1.5 PCI Express* Receiver Lane Polarity Inversion

The PCI Express* Base Specification requires polarity inversion to be supported independently by all receivers across a Link where 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 PCIe* Controllers 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".

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

18.2 Signal Description

Signal Name	Type	Description
PCIE _[24:1]_TXN **PCIE _[24:1]_TXP**	O	**PCI Express* Differential Transmit Pairs** These are the PCI Express* based outbound high-speed differential signals
PCIE_[24:1]_RXN **PCIE_[24:1]_RXP**	I	**PCI Express* Differential Receive Pairs** These are the PCI Express* based inbound high-speed differential signals
PCIE_RCOMP2 **PCIE_RCOMP5**	I	**PCI Express* PHY Impedance Compensation Inputs**
PCIE_LINK_DOWN	O	**PCI Express* Link Down Debug Signal** PCIe* link failure debug signal. 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.

18.3 I/O Signal Planes and States

Table 60. Power Plane and States for PCI Express* Signals

Signal Name	Type	Power Plane	During Reset[2]	Immediately After Reset[2]	S3/S4/S5	Deep Sx
PCIE _[24:1]_TXN	O	Primary	Internal Pull-down	Internal	Internal Pull-down	OFF

continued...

Signal Name	Type	Power Plane	During Reset[2]	Immediately After Reset[2]	S3/S4/S5	Deep Sx
PCIE _[24:1]_TXP				Pull-down		
PCIE _[24:1]_RXN **PCIE _[24:1]_RXP**	I	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
PCIE_RCOMP2 **PCIE_RCOMP5**	I	Primary	Undriven	Undriven	Undriven	OFF

Notes: 1. PCIE _[24:1]_RXN/PCIE _[24:1]_RXP pins transition from un-driven to Internal Pull-down during Reset.
2. Reset reference for primary well pins is PCH_RSMRST#.

18.4 PCI Express* Root Port Support Feature Details

Table 61. PCI Express* Root Port Feature Details

Product	Max Transfer Rate	Max Devices (Root Ports)	Max Lanes	PCIe Gen Type	Encoding	Transfer Rate (MT/s)	Theoretical Max Bandwidth (GB/s)		
							x1	x2	x4
PCH-S	16 GT/s (Gen4)	14 [2]	24	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
				4	128b/130b	16000	1.97	3.94	7.88

Notes: 1. Theoretical Maximum Bandwidth (GB/s) = ((Transfer Rate * Encoding * # PCIe Lanes) /8)/1000
- Gen4 with 4 PCIe* Lanes Example = ((16000 * 128/130* 4)/8)/1000 = 7.88 GB/s

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

Figure 12. Supported PCI Express* Link Configurations

PCH-S	FIA-2																				FIA-3			
Flex I/O Lanes	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
PCIe Controllers	1				2				3				4				5				6			
PCIe Max Rate	Gen4				Gen4				Gen4				Gen4				Gen4				Gen4			
PCIe* Lanes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
PCIe Configurations (Bi-Furcation) [3]	1px4				1px4				1px4				1px4				1px4				1px4			
	1px4(LR)				1px4(LR)				1px4(LR)				1px4(LR)				1px4(LR)				1px4(LR)			
	2px2				2px2				2px2				2px2				2px2				2px2			
	2px2(LR)				2px2(LR)				2px2(LR)				2px2(LR)				2px2(LR)				2px2(LR)			
	1px2+2px1				1px2+2px1				1px2+2px1				1px2+2px1				1px2+2px1				1px2+2px1			
	2px1+1px2 [2]				2px1+1px2 [2]				2px1+1px2 [2]				2px1+1px2 [2]				2px1+1px2 [2]				2px1+1px2 [2]			
	4px1				4px1				4px1				4px1				4px1				4px1			
Logical Link Lanes	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Assigned Root Ports	RP1				RP5				RP9				RP13				RP17				RP21			
	RP1				RP5				RP9				RP13				RP17				RP21			
	RP1		RP3		RP5		RP7		RP9		RP11		RP13		RP15		RP17		RP19		RP21		RP23	
	RP3		RP1		RP7		RP5		RP11		RP9		RP15		RP13		RP19		RP17		RP23		RP21	
	RP1		RP3	RP4	RP5		RP7	RP8	RP9		RP11	RP12	RP13		RP15	RP16	RP17		RP19	RP20	RP21		RP23	RP24
	RP4	RP3	RP1		RP8	RP7	RP5		RP12	RP11	RP9		RP16	RP15	RP13		RP20	RP19	RP17		RP24	RP23	RP21	
	RP1	RP2	RP3	RP4	RP5	RP6	RP7	RP8	RP9	RP10	RP11	RP12	RP13	RP14	RP15	RP16	RP17	RP18	RP19	RP20	RP21	RP22	RP23	RP24
Bus - Dev - Func (BDF) Assignments [1]	RP	Bus	Dev	Func	RP	Bus	Dev	Func	RP	Bus	Dev	Func	RP	Bus	Dev	Func	RP	Bus	Dev	Func	RP	Bus	Dev	Func
	1	80h	1Ch	0h	5	80h	1Ch	4h	9	80h	1Dh	0h	13	80h	1Dh	4h	17	80h	1Bh	0h	21	80h	1Bh	4h
	2	80h	1Ch	1h	6	80h	1Ch	5h	10	80h	1Dh	1h	14	80h	1Dh	5h	18	80h	1Bh	1h	22	80h	1Bh	5h
	3	80h	1Ch	2h	7	80h	1Ch	6h	11	80h	1Dh	2h	15	80h	1Dh	6h	19	80h	1Bh	2h	23	80h	1Bh	6h
	4	80h	1Ch	3h	8	80h	1Ch	7h	12	80h	1Dh	3h	16	80h	1Dh	7h	20	80h	1Bh	3h	24	80h	1Bh	7h

NOTES

1. Device (BDF) groupings have multiple functions, the lowest active Root Port within the Device (BDF) grouping will always be assigned Function 0 while any remaining active Root Port within the Device (BDF) grouping will be assigned their mapped Function # as shown.
2. 2px1+1px2 is based off selecting 1px2+2px1 with Lane Reversal Enabled
3. Reduced Root Port width configurations, within Bi-Furcation configurations, are supported (example: x2 PCIe End Point Device populated in a PCIe Controller set as 1px4 will result in a 1px2 PCIe Root Port configuration or x1 PCIe End Point Device populated in a PCIe Controller set as 1px4 will result in a 1px1 PCIe Root Port configuration).
4. FIA = Flex-IO Adapter
5. The PCIe* Link Configuration support will vary depending on the SKU. Refer to the SKU details covered in the Introduction on page 13
6. LR = Lane Reversal
7. PCIe* Configuration (#p) x (#) = (Number of PCIe* Root Ports) x (Number of Data Lane Pairs per PCIe Root Port)
8. RP# refers to a specific PCI Express* Root Port #; for example RP3 = PCI Express* Root Port 3
9. A PCIe* Lane is composed of a single pair of Transmit (TX) and Receive (RX) differential pairs. 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).
10. The PCIe* Lanes can be configured independently from one another but the max number of configured Root Ports (Devices) must not be exceeded
11. Unidentified lanes within a PCIe* Link Configuration are disabled but their physical lanes are used for the identified Root Port

19.0 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 62. 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

19.1 Signal Description

Signal Name	Type	Description
RTCX1	I	**Crystal Input 1:** This signal is connected to the 32.768 kHz crystal (max 50 kohm 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 50 kohm 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
		Note: 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.
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. SRTCRST# and RTCRST# should not be shorted together.

19.2 I/O Signal Planes and States

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

20.0 Serial ATA (SATA)

The SATA controller support two modes of operation, AHCI mode using memory space and RAID mode. The SATA controller does not support IDE legacy mode using I/O space. Therefore, AHCI software is required. The SATA controller supports the Serial ATA Specification, Revision 3.2.

Not all functions and capabilities may be available on all SKUs. Refer to PCH I/O Capabilities table and PCH SKUs table for details on feature availability.

Table 63. 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 64. 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

20.1 Functional Description

The SATA host controller supports AHCI or RAID mode.

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

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

20.1.1 Features Supported

The 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 SATA controller register .

The SATA supports the following features:

- Port Multiplier:
  - The SATA controller may optionally support command-based switching Port Multipliers.

    NOTE

    BIOS must clear this bit if Port Multipliers are not supported.

- Host Initiated Loopback Mode:
  - The need to use PxCMD.CLO to clear the internal BSY bit before setting the PxCMD.ST to '1', as there is no power-up register FIS from device to clear the BSY bit.
  - Only one CI bit can be set per each Loopback session.
  - Only one PRD can be used in each CI.
  - Only PRD size of 256B is supported.
  - Only CL.CFL=0 is supported
  - Need to set the CL.P bit for the CI, to allow PRD prefetch, without waiting for device FIS.
  - Need to clear the PxCMD.ST bit after the loopback session done and set it again to start another session.

The SATA controller does **not** support:

- FIS Based Switching
- IDE mode or combination mode
- Cold Presence Detect

20.1.2 SATA 6 Gb/s Support

The SATA controller is SATA 6 Gb/s capable and supports 6 Gb/s transfers with all capable SATA devices. The SATA controller also supports SATA 1.5 Gb/s and 3 Gb/s transfer capabilities.

20.1.3 Hot Plug Operation

The 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.

20.1.4 Intel® Rapid Storage Technology (Intel® RST)

The SATA controller provides support for Intel® Rapid Storage Technology, providing both AHCI and integrated RAID functionality. 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 SATA controller.

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 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 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.

20.1.5 Power Management Operation

Power management of the 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 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 host 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

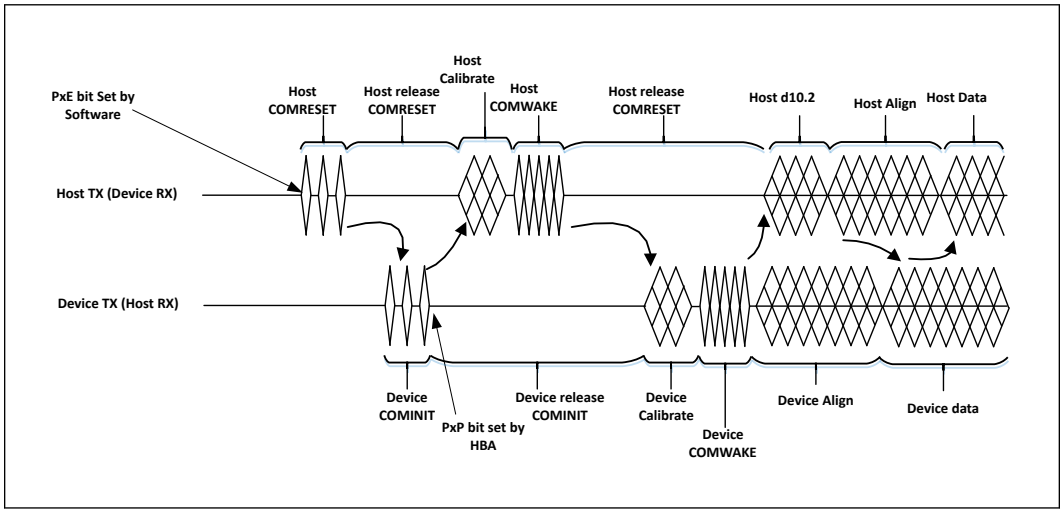
When low power feature is enabled, the Intel® SATA controller may power off PLLs or OOB detection circuitry while in the Slumber link power state. As a result, a device initiated wake may not be recognized by the host. For example, when the low power feature is enabled it can prevent a Zero Power ODD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

20.1.6 SATA Device Presence

The flow used to indicate SATA device presence is shown in the Figure below. The 'PxE' bit refers to bits, depending on the port being checked and the 'PxP' bits refer to the 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 13. Port Enable/Device Present Bits Flow



20.1.7 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.

20.1.8 Advanced Host Controller Interface (AHCI) Operation

The 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 host—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 host/device designation for SATA devices—each device is treated as a host—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 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.

20.1.9 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.

This controller supports multiple-activity LEDs that show per-drive status information. Typical LEDs per drive include device activity, fault (to indicate a failure), and locate (to show an operator the bay to remove/insert a drive from/into).

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 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 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 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 Host Controller does not ensure that it will cause the target SGPIO device or controller to be reset. Software is responsible to keep the 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 section 25.3.7.3, LED Message Type. It is the responsibility of software to ensure that the content of the message format is correct. If the message type is not programmed as 'LED' for this controller, the controller will not take any action to update its LEDs.

Data Size (DSIZE) specifies the date size in bytes. If the message (enclosure services command) has a data buffer that is associated with it that is transferred, the size of that data buffer is specified in this field. If there is no separate data buffer, this field shall have a value of '0'. The data directly follows the message in the message buffer. This value should always be '0'.

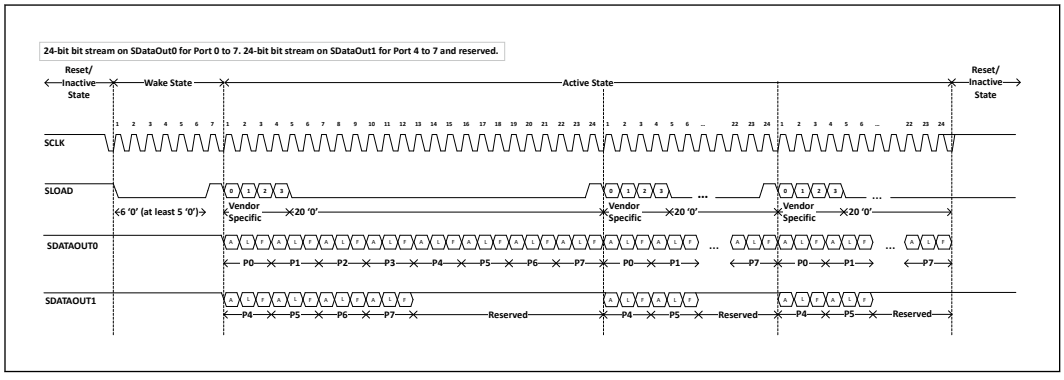
Message Size (MSIZE) specifies the size of the message in bytes . The message size does not include the one Dword header. A value of '0' is invalid. The message size is always 4 bytes.

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) register. The LEDs shall retain their values until there is a following update for that particular slot.

SGPIO Waveform

Figure 14. Serial Data Transmitted over SGPIO Interface



20.2 Signal Description

Signal Name	Type	Description
GPP_E04/**SATA_DEVSLP0**	I or O	**Serial ATA Port [0] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state. Design Constraint: no external Pull-up or Pull-down termination required when used as DEVSLP. *Note:* This pin can be mapped to SATA Port 0.
GPP_E05/**SATA_DEVSLP1**	I or O	**Serial ATA Port [1] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri- state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state. Design Constraint: no external Pull-up or Pull-down termination required when used as DEVSLP. *Note:* This pin can be mapped to SATA Port 1.
GPP_E06/**SATA_DEVSLP2**	I or O	**Serial ATA Port [2] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state. Design Constraint: no external Pull-up or Pull-down termination required when used as DEVSLP. *Note:* This pin can be mapped to SATA Port 2.
GPP_F05/**SATA_DEVSLP3**	I or O	**Serial ATA Port [3] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state. Design Constraint: no external Pull-up or Pull-down termination required when used as DEVSLP. *Note:* This pin can be mapped to SATA Port 3.
GPP_F06/**SATA_DEVSLP4**	I or O	**Serial ATA Port [4] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state. Design Constraint: no external Pull-up or Pull-down termination required when used as DEVSLP. *Note:* This pin can be mapped to SATA Port 4.
GPP_F07/**SATA_DEVSLP5**	I or O	**Serial ATA Port [5] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state. Design Constraint: no external Pull-up or Pull-down termination required when used as DEVSLP. *Note:* This pin can be mapped to SATA Port 5.
GPP_F08/**SATA_DEVSLP6**	I or O	**Serial ATA Port [6] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state. Design Constraint: no external Pull-up or Pull-down termination required when used as DEVSLP.

continued...

Signal Name	Type	Description
		Note: This pin can be mapped to SATA Port 6.
GPP_F09/**SATA_DEVSLP7**	I or O	**Serial ATA Port [7] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state. Design Constraint: no external Pull-up or Pull-down termination required when used as DEVSLP. *Note:* This pin can be mapped to SATA Port 7.
PCIE_13_TXN/**SATA_0_TXN** PCIE_13_TXP/**SATA_0_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.
PCIE_13_RXN/**SATA_0_RXN** PCIE_13_RXP/**SATA_0_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.
PCIE_14_TXN/**SATA_1_TXN** PCIE_14_TXP/**SATA_1_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.
PCIE_14_RXN/**SATA_1_RXN** PCIE_14_RXP/**SATA_1_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.
PCIE_15_TXP/**SATA_2_TXP**/GbE_0C_TXP PCIE_15_TXN/**SATA_2_TXN**/GbE_0C_TXN	O	**Serial ATA Differential Transmit Pair 2**: These outbound SATA Port 2 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE_15_RXP/**SATA_2_RXP**/GbE_0C_RXP PCIE_15_RXN/**SATA_2_RXN**/GbE_0C_RXN	I	**Serial ATA Differential Receive Pair 2**: These inbound SATA Port 2 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE_16_TXP/**SATA_3_TXP** PCIE_16_TXN/**SATA_3_TXN**	O	**Serial ATA Differential Transmit Pair 3**: These outbound SATA Port 3 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE_16_RXP/**SATA_3_RXP** PCIE_16_RXN/**SATA_3_RXN**	I	**Serial ATA Differential Receive Pair 3**: These inbound SATA Port 3 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE_17_TXP/**SATA_4_TXP** PCIE_17_TXN/**SATA_4_TXN**	O	**Serial ATA Differential Transmit Pair 4**: These outbound SATA Port 4 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE_17_RXP/**SATA_4_RXP** PCIE_17_RXN/**SATA_4_RXN**	I	**Serial ATA Differential Receive Pair 4**: These inbound SATA Port 4 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE_18_TXP/**SATA_5_TXP** PCIE_18_TXN/**SATA_5_TXN**	O	**Serial ATA Differential Transmit Pair 5**: These outbound SATA Port 5 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE_18_RXP/**SATA_5_RXP** PCIE_18_RXN/**SATA_5_RXN**	I	**Serial ATA Differential Receive Pair 5**: These inbound SATA Port 5 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE_19_TXP/**SATA_6_TXP** PCIE_19_TXN/**SATA_6_TXN**	O	**Serial ATA Differential Transmit Pair 6**: These outbound SATA Port 6 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE_19_RXP/**SATA_6_RXP** PCIE_19_RXN/**SATA_6_RXN**	I	**Serial ATA Differential Receive Pair 6**: These inbound SATA Port 6 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE_20_TXP/**SATA_7_TXP** PCIE_20_TXN/**SATA_7_TXN**	O	**Serial ATA Differential Transmit Pair 7**: These outbound SATA Port 7 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE_20_RXP/**SATA_7_RXP** PCIE_20_RXN/**SATA_7_RXN**	I	**Serial ATA Differential Receive Pair 7**: These inbound SATA Port 7 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.

continued...

Signal Name	Type	Description
GPP_E00/**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(SATA [bit0] or PCIe* [bit1] combo lanes selection) depends on soft-strap.
GPP_E01/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. *Note:* This default use of this pin is GPP_E1.Pin defaults to Native mode as SATAXPCIE1 (SATA [bit0] or PCIe* [bit1] combo lanes selection) depends on soft-strap.
GPP_E02/SATAXPCIE2/**SATAGP2**	I	**Serial ATA Port [2] General Purpose Inputs**: When configured as SATAGP2, this is an input pin that is used as an interlock switch status indicator for SATA Port 2. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open. *Note:* This default use of this pin is GPP_E2.Pin defaults to Native mode as SATAXPCIE2 (SATA [bit0] or PCIe* [bit1] combo lanes selection) depends on soft-strap.
GPP_F00/SATAXPCIE3/**SATAGP3**	I	**Serial ATA Port [3] General Purpose Inputs**: When configured as SATAGP3, this is an input pin that is used as an interlock switch status indicator for SATA Port 3. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open. *Note:* This default use of this pin is GPP_F0.Pin defaults to Native mode as SATAXPCIE3 (SATA [bit0] or PCIe* [bit1] combo lanes selection) depends on soft-strap.
GPP_F01/SATAXPCIE4/**SATAGP4**	I	**Serial ATA Port [4] General Purpose Inputs**: When configured as SATAGP4, this is an input pin that is used as an interlock switch status indicator for SATA Port 4. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open. *Note:* This default use of this pin is GPP_F1.Pin defaults to Native mode as SATAXPCIE4 (SATA [bit0] or PCIe* [bit1] combo lanes selection) depends on soft-strap.
GPP_F02/SATAXPCIE5/**SATAGP5**	I	**Serial ATA Port [5] General Purpose Inputs**: When configured as SATAGP5, this is an input pin that is used as an interlock switch status indicator for SATA Port 1. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open. *Note:* This default use of this pin is GPP_F2.Pin defaults to Native mode as SATAXPCIE5 (SATA [bit0] or PCIe* [bit1] combo lanes selection) depends on soft-strap.
GPP_F03/SATAXPCIE6/**SATAGP6**	I	**Serial ATA Port [6] General Purpose Inputs**: When configured as SATAGP6, this is an input pin that is used as an interlock switch status indicator for SATA Port 6. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open. *Note:* This default use of this pin is GPP_F3.Pin defaults to Native mode as SATAXPCIE6 (SATA [bit0] or PCIe* [bit1] combo lanes selection) depends on soft-strap.
GPP_F04/SATAXPCIE7/**SATAGP7**	I	**Serial ATA Port [7] General Purpose Inputs**: When configured as SATAGP7, this is an input pin that is used as an interlock switch status indicator for SATA Port 1. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open. *Note:* This default use of this pin is GPP_F4.Pin defaults to Native mode as SATAXPCIE7 (SATA [bit0] or PCIe* [bit1] combo lanes selection) depends on soft-strap.

continued...

Signal Name	Type	Description
GPP_E08/**SATALED#**	I or O	**Serial ATA LED**: This signal is an open-drain output pin driven during SATA command activity. It is to be connected to external circuitry that can provide the current to drive a platform LED. When active, the LED is on. When tri-stated, the LED is off. *Note:* An external Pull-up resistor to VCC1P8 is required.
GPP_F13/**SATA_SDATAOUT0**	O	**Serial ATA GPIO Data Output 0** :Enclosure Management SGPIO Dataout 0 to indicate the drive status
GPP_F12/**SATA_SDATAOUT1**	O	**Serial ATA GPIO Data Output 1** :Enclosure Management SGPIO Dataout 1 to indicate the drive status
GPP_F11/**SATA_SLOAD**	I or O	**Serial ATA GPIO Load Signal** :Enclosure Management SGPIO Load to indicate either the start or end of a bit stream.
GPP_F10/**SATA_SCLOCK**	I or O	**Serial ATA GPIO Reference Clock** :Enclosure Management SGPIO Reference Clock

20.3 Integrated Pull-Ups and Pull-Downs

Signal Name	Resistor Type
SATAXPCIE[0:7]	Internal Pull-up
SATA_[0:7]_RXN	Internal Termination
SATA_[0:7]_RXP	Internal Termination
SATA_[0:7]_TXN	Internal Termination
SATA_[0:7]_TXP	Internal Termination
SATAGP[0:7]	Internal Termination
SATA_DEVSLP[0:7]	External Pull-up
SATA_SDATAOUT[0:1]	External Pull-up
SATA_SLOAD	External Pull-up
SATA_SCLOCK	External Pull-up
Note: Internal Pull-Up Resistors are 20 kohm ± 30% unless specified.	

20.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[3]	Immediately after Reset[3]	S4/S5
SATA_[0:7]_TXN SATA_[0:7]_TXP SATA_[0:7]_RXN SATA_[0:7]_RXP	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down
SATALED#	Primary	Undriven	Undriven	Undriven
SATA_DEVSLP[0:7][1]	Primary	Undriven	Undriven	Undriven
SATAGP[0:7][2]	Primary	Undriven	Undriven	Undriven
SATAXPCIE[0:7][2]	Primary	Internal Pull-up	Internal Pull-up	Undriven
SATA_SDATAOUT[0:1]	Primary	Undriven	Undriven	Undriven

continued...

Signal Name	Power Plane	During Reset[3]	Immediately after Reset[3]	S4/S5
SATA_SLOAD	Primary	Undriven	Undriven	Undriven
SATA_SCLOCK	Primary	Undriven	Undriven	Undriven

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 S4/S5 reflects assumption that GPIO Use Select register was programmed to native mode functionality. If the GPIO Use Select register is programmed to GPIO mode, refer to Intel® 800 Series Chipset Family PCH - GPIO implementation summary for the pin state during reset and pin state immediately after reset.
2. Pin defaults to Native mode as SATAXPCIEx depends on soft-strap.
3. Reset reference for primary well pins is PCH_RSMRST#.

21.0 Serial Peripheral Interface (SPI)

The PCH provides Serial Peripheral Interfaces (SPI) to connect up to two flash devices. 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 supports either 1.8 V or 3.3V.

Table 65. Acronyms

Acronyms	Description
CLK	Clock
CS	Chip Select
FCBA	Flash Component Base Address
FLA	Flash Linear Address
FMBA	Flash Controller Base Address
FPSBA	Flash Processor Strap Base Address
FRBA	Flash Region Base Address
MDTBA	MIP Descriptor Table Base Address
MISO	Terminology to indicate signal direction: input to the host, output from the device
MOSI	Terminology to indicate signal direction: output from the host, input to the device
TPM	Trusted Platform Module

21.1 Functional Description

21.1.1 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. The PCH drives the SPI0 interface clock at either 14 MHz, 25 MHz, 33 MHz and 50 MHz and will function with SPI flash/TPM devices that support at least one of these frequencies.

A SPI0 flash device supporting SFDP (Serial Flash Discovery Parameter) is required for all 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 for all SKUs of the PCH. Non-Descriptor Mode is not supported.

- **SPI0 Flash Regions**

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

Table 66. SPI0 Flash Regions

Region	Content
0	Flash Descriptor
1	BIOS
2	Intel ®CSME
3	GbE - Location for Integrated LAN firmware and MAC address
4	PDR - Platform Data Region
8	EC - Embedded Controller
10	Intel® Silicon Security Engine

Only four controllers 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 67. 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

continued...

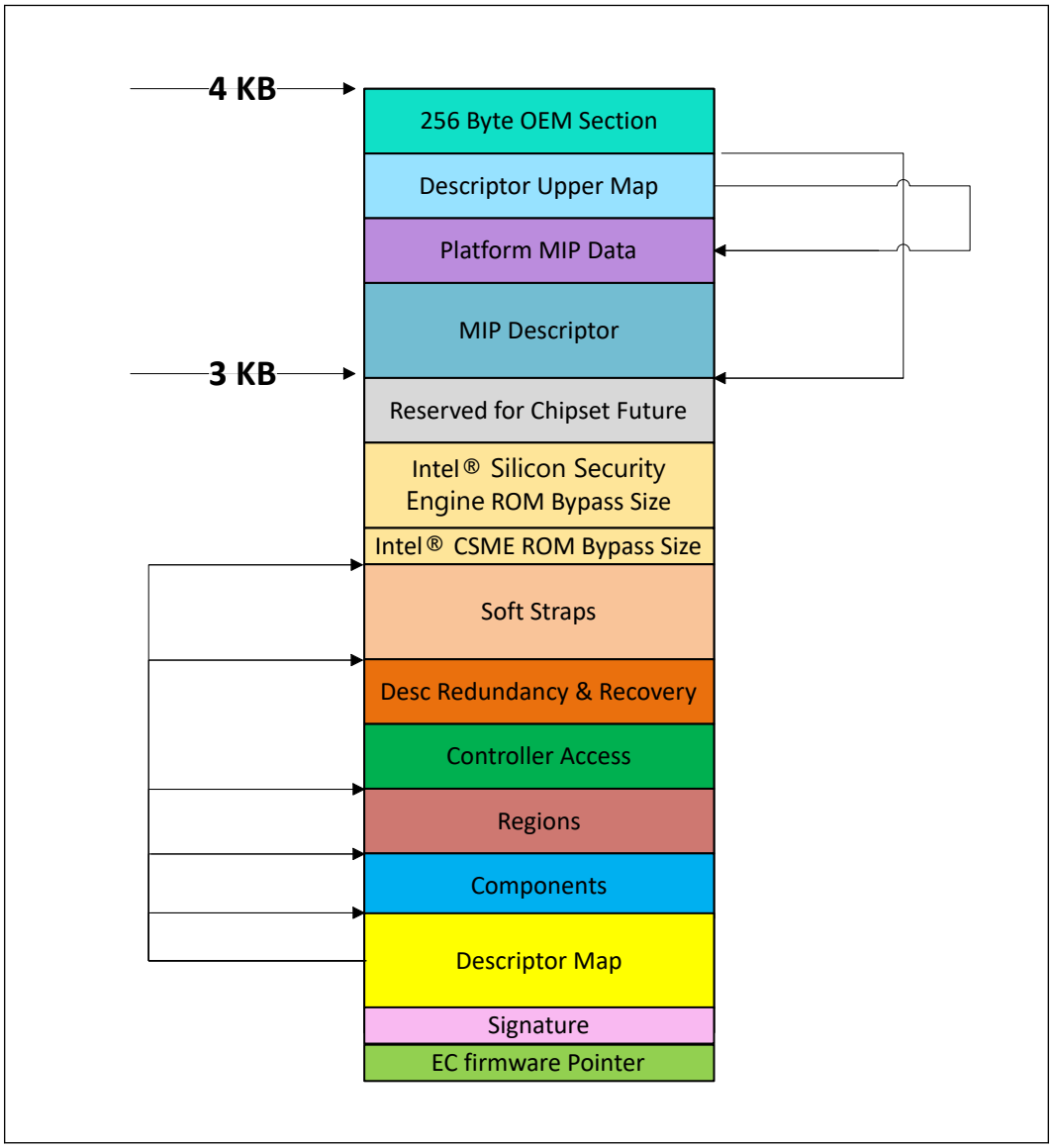
Region	Size with 4 KB Blocks	Size with 8 KB Blocks	Size with 64 KB Blocks
BIOS	Varies by Platform	Varies by Platform	Varies by Platform
Intel® CSME	Varies by Platform	Varies by Platform	Varies by Platform
EC	Varies by Platform	Varies by Platform	Varies by Platform
PDR	Varies by Platform	Varies by Platform	Varies by Platform
Intel® CSME Data	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. It requires its own discrete erase block, so it may need greater than 4 KB of flash space depending on the flash architecture that is on the target system. Two additional redundant back-ups of the Flash Descriptor have been added for data resilience.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 fifteen sections as shown in the figure below:

Figure 15. Flash Descriptor Regions



- EC Firmware Pointer is located in the first 16 bytes of the Descriptor and contains the address location for EC flash region. The format for the EC Firmware Pointer address is dependent on EC vendors/OEM implementation of this field.
- The Flash signature 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 lower five descriptor sections as well as the size of each.
- The Component section has information about the SPI flash part(s) the system. It includes the number of components, density of each component, read, write and erase frequencies and invalid instructions.

- The Region section defines the base and the limit of the BIOS, IFWI, GbE, Platform Data Region (PDR- Optional), Embedded Controller (EC- Optional) regions as well as their size.
- The Processor and PCH soft strap sections contain configurable parameters.
- The Reserved region is for future chipset usage.
- The Descriptor Upper Map determines the length and base address of the Intel ® CSME VSCC Table.
- The Intel ® CSME VSCC Table holds the JEDEC ID and the ME VSCC information for all the SPI Flash part(s) supported by the NVM image. BIOS and GbE write and erase capabilities depend on VSCC0 and VSCC1 registers in SPIBAR memory space.
- OEM Section is 256 bytes reserved at the top of the Flash Descriptor for use by OEM.

Descriptor Controller Region

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

Table 68. Region Access Control Table

Controller Read/Write Access				
Region	**Processor and BIOS**	**Intel® CSME**	**GbE Controller**	**EC**
Descriptor (0)	Read Only	Read Only	Not Accessible	Not Accessible
BIOS (1)	processor / BIOS can always read from and write to BIOS region prior to EOP	Not Accessible	Not Accessible	Not Accessible
Intel® CSME (2)	Read/Write (BIOS Only)	Intel® CSME can always read from and write to firmware region	Not Accessible	Not Accessible
Gigabit Ethernet (3)	Not Accessible	Read/Write	GbE software can always read from and write to GbE region	Not Accessible
PDR (4)	Not Accessible	Not Accessible	Not Accessible	Not Accessible
EC (8)	Read/Write	Not Accessible	Not Accessible	EC can always read from and write to EC region.
Intel® CSME Data (15)	Not Accessible	Read/Write	Not Accessible	Not Accessible

Notes:
- The Region Access values listed above represent post manufacturing configuration only.
- Descriptor and PDR region is not a Controller, so they will not have Controller R/W access.
- Descriptor should NOT have write access by any Controller in production systems.
- PDR region should only have read and/or write access by processor/Host. GbE and Intel® CSME should NOT have access to PDR region.

Table 69. Flash Descriptor Processor Complex Soft Strap

Region Name	Starting Address
Signature	10h
Component FCBA	30h
Regions FRBA	40h
Controllers FMBA	80h
PCH Straps FPSBA	100h
Desc Redundancy & Recovery	320h
MDTBA	C00h
SOC-S PMC Straps	C14h
IOE PMC Straps	C6Ch
IOE Soft Straps	CACh
Processor Straps	CECh
Intel® CSME Straps	D9Ch

Flash Access

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

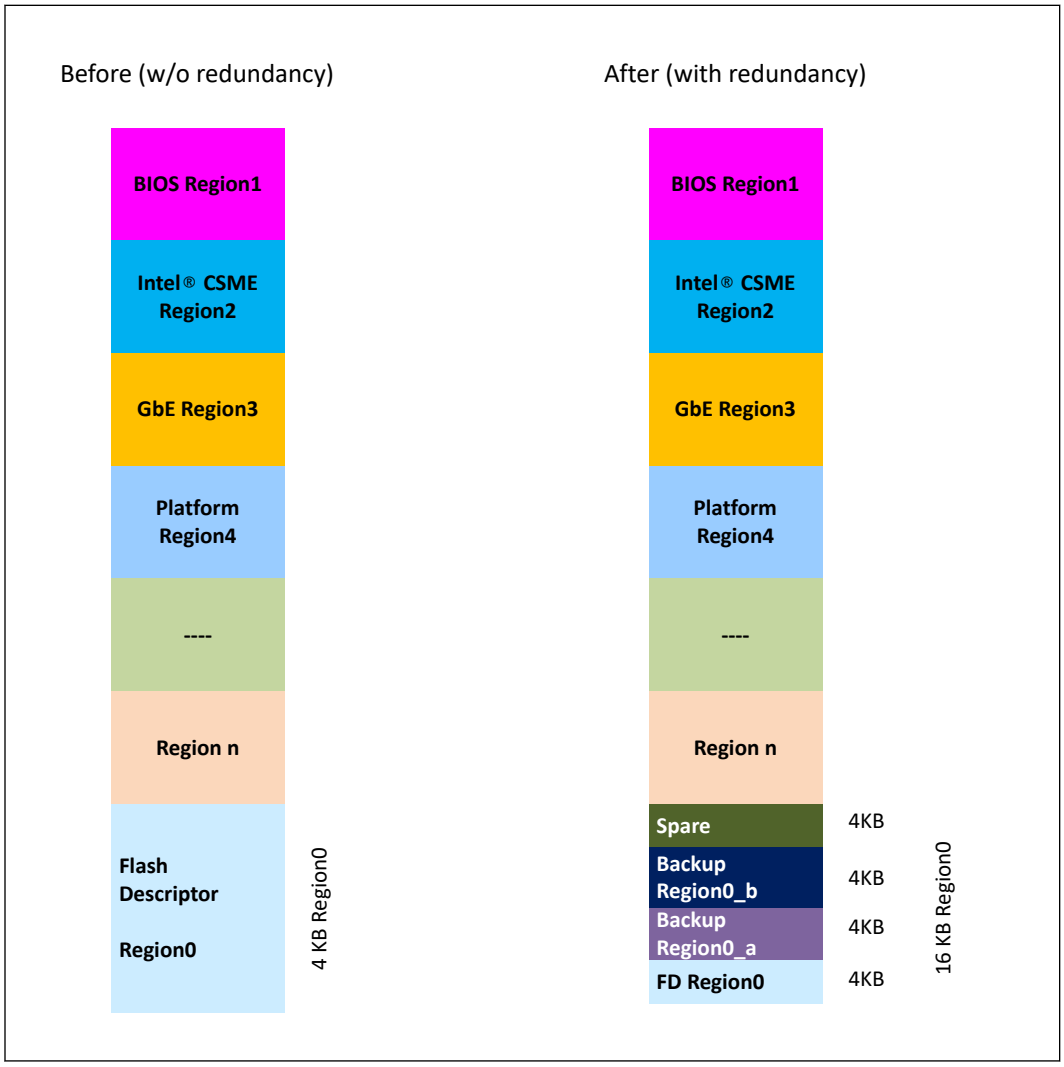
- **Direct Access**
  - Controllers are allowed to do direct read only of their primary region
    - Gigabit Ethernet region can only be directly accessed by the Gigabit Ethernet controller. Gigabit Ethernet software must use Program Registers to access the Gigabit Ethernet region.
  - Controller's Host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

    Direct Access Security
  - Requester ID of the device must match that of the primary Requester ID in the Controller 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 Controller
- **Program Register Access**
  - Program Register Accesses are not allowed to cross a 4 KB boundary and cannot 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 Controllers can access the registers

Flash Descriptor Redundancy and Recovery

In order to provide descriptor redundancy and recovery, SPI flash controller uses two 4 KB spaces or regions as the backup descriptor regions. Each backup descriptor region size is 4 KB.

Figure 16. Flash Descriptor Redundancy



In the main and backup descriptor regions, the following fields are defined for the descriptor integrity check and recovery. Before SPI controller reads the descriptor, it

- Reads Main Descriptor Region and calculates SHA-256 hash.
- Reads Active Backup Descriptor Region and calculates hash.
- Compares each hash result with the hash in that region.
- Takes action based on result and policy byte (in Main Descriptor).

RPMC Configuration

Intel Replay Protection Monotonic Counter (RPMC) is a capability providing Anti-Replay Protection using Monotonic Counters inside SPI Flash. Intel RPMC is a critical security feature designed to protect the SPI part of Intel platforms from unauthorized write operations. This innovative technology acts as a robust defense mechanism, ensuring that only authorized write operations are permitted, thus preventing any unauthorized access to the SPI.

RPMC protection relies on:

- Special RPMC HW and logic inside the SPI Flash.
- Intel CSME FW support that utilizes RPMC capabilities within Flash.

At the core of RPMC's functionality lies the concept of the session key.

The session key is a cryptographic key derived from several factors residing on the PCH. These factors are carefully selected and stored upon provisioning RPMC to the SPI part. The session key serves as a means of authenticating each incoming write message to the SPI. When an authorized operation is initiated, the session key is used to verify the legitimacy of the request. If the session key does not match the expected value, the SPI part will reject the request, effectively blocking malicious or unauthorized write operations.

Furthermore, the session key also extends its protective shield to cover a specific set of sensitive read messages. This holistic approach ensures that not only write operations but also read operations involving sensitive data are monitored and authenticated, enhancing the overall security of the system.

Two features of RPMC can be enabled:

- RPMC will be enabled on platforms with RPMC SPI. During Intel End of Manufacturing the PCH will be bound with RPMC SPI
- When SPI is replaced, re-binding between the new RPMC SPI and thePCH will happen automatically on first boot.

Monotonic Counters

Monotonic counters are counters on the SPI Flash maintained by Intel CSME FW. SPI Flash has a set of four 32-bit monotonic counters, where Intel CSME FW uses two of these counters. Intel CSME FW ensures FW write operations will not exceed SPI RPMC monotonic counter increment rate specified by RPMC HW during platform lifetime supported by Intel. Reading and incrementing the counters in the Flash is done using authenticated commands with a key known to both: SPI Flash and Intel® CSME FW

Binding at End of Manufacturing (EOM)

RPMC Binding pairs between SPI Flash and the PCH by provisioning the Binding key produced by the PCH into SPI Flash. This pairing is done as part of the EOM flow which usually takes place at the manufacturing line.

In conclusion, Intel RPMC, with its Replay Monotonic Counter and session key mechanism, stands as a powerful safeguard against unauthorized write operations and unauthorized access to sensitive data in the SPI part. This robust security feature, derived from the session key, adds an additional layer of protection to Intel platforms, making them more resilient against potential threats and ensuring the integrity and confidentiality of the data stored in the SPI.

21.1.2 SPI0 Support for TPM

The PCHSPI0 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 soft strap. 20 MHz is the reset default, a valid 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 interrupt configurable GPIO pin. Thus, TPM interrupt is completely independent from the SPI0 controller.

21.2 Signal Description

Signal Name	Type	Description
SPI0_CLK	O	**SPI0 Clock**: SPI clock signal for the common flash/TPM interface. Supports upto maximum of 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 Host OUT Device IN**: Defaults as a data output pin for the 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 Host IN Device OUT**: Defaults as a data input pin for the 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.

21.3 Integrated Pull-Ups and Pull-Downs

Signal	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

Above resistor type is dynamic state controlled by the SPI controller. The internal Pull-up is disabled when PCH_RSMRST# is asserted (during reset) and only enabled after PCH_RSMRST# de-assertion.

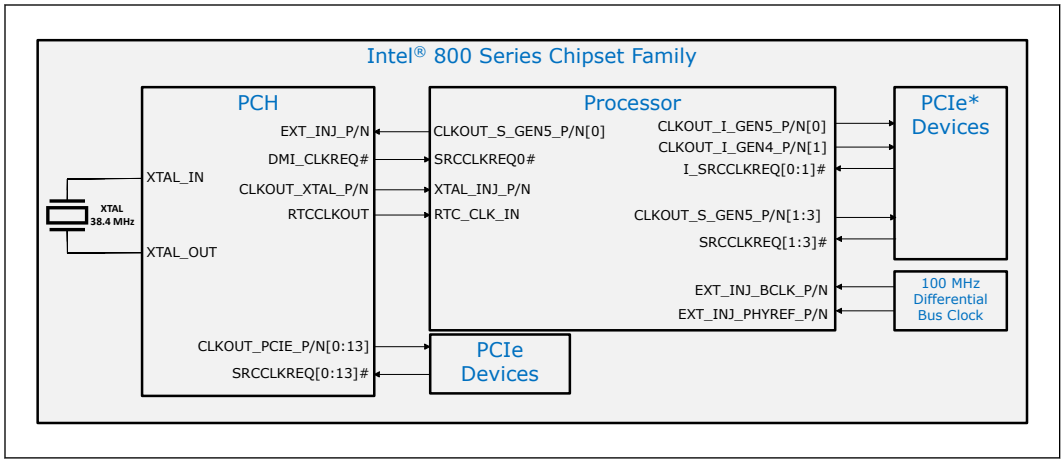
21.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5
SPI0_CLK	Primary	Internal Pull-down	Driven Low	Driven Low
SPI0_MOSI	Primary	Hi-Z	Internal Pull-up , then Driven Low	Driven Low
SPI0_MISO	Primary	Hi-Z	Internal Pull-up	Internal Pull-up
SPI0_CS[2:0]#	Primary	Internal Pull-down	Driven High	Driven High
SPI0_IO[3:2]	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up
Note: 1. During reset refers to when PCH_RSMRST# is asserted.				

22.0 System Clocks

22.1 Integrated Clock Controller (ICC)

Figure 17. ICC Diagram



22.1.1 Signal Descriptions

Table 70. Signal Descriptions

Signal Name	Type	SSC Capable	Description
CLKOUT_PCIE_N0 **CLKOUT_PCIE_N1** **CLKOUT_PCIE_N2** **CLKOUT_PCIE_N3** **CLKOUT_PCIE_N4** **CLKOUT_PCIE_N5** **CLKOUT_PCIE_N6** **CLKOUT_PCIE_N7** **CLKOUT_PCIE_N8** **CLKOUT_PCIE_N9** **CLKOUT_PCIE_N10** **CLKOUT_PCIE_N11** **CLKOUT_PCIE_N12** **CLKOUT_PCIE_N13** **CLKOUT_PCIE_P0** **CLKOUT_PCIE_P1** **CLKOUT_PCIE_P2** **CLKOUT_PCIE_P3** **CLKOUT_PCIE_P4** **CLKOUT_PCIE_P5**	O	Yes	**PCI Express* Clock Output**: Serial Reference 100 MHz PCIe* specification compliant differential output clocks to PCIe* devices. CLKOUT_PCIE_P/N [13:0] is used for PCIe* Gen4 support.

continued...

Signal Name	Type	SSC Capable	Description
CLKOUT_PCIE_P6 **CLKOUT_PCIE_P7** **CLKOUT_PCIE_P8** **CLKOUT_PCIE_P9** **CLKOUT_PCIE_P10** **CLKOUT_PCIE_P11** **CLKOUT_PCIE_P12** **CLKOUT_PCIE_P13**			
GPP_D00/**SRCCLKREQ0#** GPP_D01/**SRCCLKREQ1#** GPP_D02/**SRCCLKREQ2#** GPP_D03/**SRCCLKREQ3#** GPP_D11/**SRCCLKREQ4#** GPP_D12/**SRCCLKREQ5#** GPP_D13/**SRCCLKREQ6#** GPP_D14/**SRCCLKREQ7#** GPP_H01/**SRCCLKREQ8#**/USB-C_GPP_H01 GPP_H02/**SRCCLKREQ9#**/USB-C_GPP_H02 GPP_H03/**SRCCLKREQ10#**/USB-C_GPP_H03 GPP_H04/**SRCCLKREQ11#**/USB-C_GPP_H04 GPP_E09/**SRCCLKREQ12#** GPP_E10/**SRCCLKREQ13#**	IOD		**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.
CLKOUT_XTAL_P **CLKOUT_XTAL_N**	O		38.4 MHz XTAL NS clock output to Processor.
GPP_J08/**RTCCLKOUT**	O		A form of RTC clock from the PCH to the processor which is active in S0, S0ix, and Sx.
EXT_INJ_P **EXT_INJ_N**	I		External injected differential clock input for DMI overclocking.
DMI_CLKREQ#	IOD		**Clock Request:** Clock request for DMI controller.
PCIE_EXTREFCLKN **PCIE_EXTREFCLKP**	I		External Differential Reference Clock Input - for PCIe.

Notes: 1. SSC = Spread Spectrum Clocking. Intel does not recommend changing the Plan of Record and fully validated SSC default value set in BIOS Reference Code. The SSC level must only be adjusted for debugging or testing efforts and any Non POR configuration setting used are the sole responsibility of the customer.

2. The SRCCLKREQ# signals can be configured to map to any of the PCI Express* Root Ports while using any of the CLKOUT differential pairs.

22.2 I/O Signal Pin States

Table 71. I/O Signal Pin States

Signal Name	Power Plane	During Reset[1]	Immediately After Reset[1]	S4/S5
CLKOUT_PCIE_P[0:13]	Primary	Toggling	Toggling	OFF (Gated Low)

continued...

Signal Name	Power Plane	During Reset[1]	Immediately After Reset[1]	S4/S5
CLKOUT_PCIE_N[0:13]				
SRCCLKREQ[0:13]#	Primary	Un-driven	Un-driven	Un-driven
EXT_INJ_P/N	Primary	Un-driven	Un-driven	Un-driven
1. Reset reference for primary well pin is PCH_RSMRST#.				

23.0 System Management Interface and SMLink

The PCH provides Five SMLink interfaces, SMLink0 to SMLink4. The interfaces are intended for system management and are controlled by the Intel® CSME. Refer to System Management on page 176 for more detail.

Table 72. Acronyms

Acronyms	Description
BMC	Baseboard Management Controller
EC	Embedded Controller

23.1 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 SMT3_EN 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.

23.1.1 Integrated USB-C* Usage

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

SMLINK1 uses controller 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 the processor 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 at one side or both side of the system,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 PMC will handle the Re-timer and PD controller, but if the PD controller has three or more I^2C ports then 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.

23.2 Signal Description

Signal Name	Type	Description
GPP_D09/**SML0CLK**	I/OD	**System Management Link 0 Clock**: SMLink0 is used between PCH's LAN controller and the PHY. External Pull-up resistor required.
GPP_D10/**SML0DATA**	I/OD	**System Management Link 0 Data**: SMLink0 is used between PCH's LAN controller and the PHY.. External Pull-up resistor required.
GPP_C05/**SML0ALERT#**/ USB-C_GPP_C05	I/OD	**System Management 0 Alert**: SMLink0 is used between PCH's LAN controller and the PHY. External Pull-up resistor required.
GPP_D04/**SML1CLK**	I/OD	**System Management Link 1 Clock**: SMBus link to optional Embedded Controller or BMC. External Pull-up resistor required.
GPP_D15/**SML1DATA**	I/OD	**System Management Link 1 Data**: SMBus link to optional Embedded Controller or BMC. External Pull-up resistor required.
GPP_B21/**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. SMLink4 should be used for USB-C* usage. External Pull-up resistor is required on this pin.
GPP_H09/**SML2CLK**/USB- C_GPP_H09	I/OD	**System Management Link 2 Clock**: SMBus link to external PHY. External Pull-up resistor required.
GPP_H10/**SML2DATA**/ USB-C_GPP_H10	I/OD	**System Management Link 2 Data** External Pull-up resistor required.
GPP_H11/**SML2ALERT#**/ USB-C_GPP_H11	I/OD	**System Management 2 Alert**: Alert for the SMBus controller to optional Embedded Controller or BMC. External Pull-up resistor required.
GPP_H12/**SML3CLK**/USB- C_GPP_H12	I/OD	**System Management Link 3 Clock**: SMBus link to external PHY. External Pull-up resistor required.
GPP_H13/**SML3DATA**/ USB-C_GPP_H13	I/OD	**System Management Link 3 Data** External Pull-up resistor required.
GPP_H14/**SML3ALERT#**/ USB-C_GPP_H14	I/OD	**System Management 3 Alert**: Alert for the SMBus controller to optional Embedded Controller or BMC. External Pull-up resistor required.
GPP_H15/**SML4CLK**/USB- C_GPP_H15	I/OD	**System Management Link 4 Clock**: SMBus link to PD Controllers. External Pull-up resistor required.
GPP_H16/**SML4DATA**/ USB-C_GPP_H16	I/OD	**System Management Link 4 Data** SMBus link to PD Controllers. External Pull-up resistor required.

continued...

Signal Name	Type	Description
GPP_H17/**SML4ALERT#**/ USB-C_GPP_H17	I/OD	**System Management 4 Alert**: SMBus link to PD Controllers. External Pull-up resistor required.
GPP_C04/ ISH_UART0_TXD/ **SML0BCLK**/I2C2_SCL/ USB-C_GPP_C04	I/OD	**System Management Link 0 B Clock** External Pull-up resistor required. *Note:* Alternate interface from/to same SML0 controller
GPP_C03/ ISH_UART0_RXD/ **SML0BDATA**/I2C2_SDA/ USB-C_GPP_C03	I/OD	**System Management Link 0 B Data** External Pull-up resistor required. *Note:* Alternate interface from/to same SML0 controller
GPP_E21/ ISH_UART0_CTS#/ **SML0BALERT#**	I/OD	**System Management 0 Alert**: Alert for the SMBus controller to optional Embedded Controller or BMC. External Pull-up resistor required. *Note:* Alternate interface from/to same SML0 controller
INTRUDER#	I	**Intruder Detect**
SMI#	OD	**System Management Interrupt (SMI)** - Event indication to BMC (Server)
NMI#	OD	**Non Maskable Interrupt (NMI)**- Event indication to BMC (Server)

23.3 Integrated Pull-Ups and Pull-Downs

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

23.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5
INTRUDER#	RTC	Undriven	Undriven	Undriven
SML[1:0]DATA	Primary	Undriven	Undriven	Undriven
SML[1:0]CLK	Primary	Undriven	Undriven	Undriven
SML[1:0]ALERT#	Primary	Pull-down (Internal)	Driven Low	Pull-down (Internal)
PCHHOT#	Primary	Pull-down (Internal)	Driven Low	Pull-down (Internal)
Note: 1. Reset reference for primary well pin is PCH_RSMRST# and RTC well pin is RTCRST#.				

24.0 Touch Host Controller (THC)

NOTE

THC-SPI is not POR for Intel® Core™ Ultra 200S Series Processors .

Touch Host Controller provides a standard SPI interface for PCH to connect to external touch ICs. 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 PCH

Table 73. Acronyms

Acronyms	Description
CLK	Clock
CS	Chip Select
TPM	Trusted Platform Module

24.1 Signal Description

Signal Name	Type	Description
GPP_D17/GSPI3_CLK/**THC1_SPI2_CLK**	O	**THC1_SPI2 Clock**: THC SPI2 clock output from PCH. Supports 42.67MHz.
GPP_D16/GSPI3_CS0#/ **THC1_SPI2_CS#**	O	**THC1_SPI2 Chip Select**: Used to select the touch devices if it is connected to THC1_SPI2 interface.
GPP_D18/GSPI3_MISO/**THC1_SPI2_IO0**	I/O	**THC1_SPI2_IO0**: A bidirectional signal used to support single, dual and quad mode data transfer.
GPP_D19/GSPI3_MOSI/**THC1_SPI2_IO1**	I/O	**THC1_SPI2_IO1**: A bidirectional signal used to support single, dual and quad mode data transfer.
GPP_D20/UART3_RXD/**THC1_SPI2_IO2**	I/O	**THC1_SPI2_IO2**: A bidirectional signal used to support single, dual and quad mode data transfer.
GPP_D21/UART3_TXD/**THC1_SPI2_IO3**	I/O	**THC1_SPI2_IO3**: A bidirectional signal used to support single, dual and quad mode data transfer.
GPP_D22/UART3_RTS#/ **THC1_SPI2_RST#**	O	**THC1 SPI2 Reset:** THC1_SPI2 Reset signal from Touch host controller.
GPP_D23/UART3_CTS#/ **THC1_SPI2_INT#**	I	**THC1 SPI2 interrupt:** THC1_SPI2 Interrupt signal.

24.2 Integrated Pull-Ups and Pull-Downs

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

NOTE

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

24.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset[1]	S4/S5
THC1_SPI2_CLK	Primary	Undriven	Undriven	Undriven
THC1_SPI2_CS#	Primary	Undriven	Undriven	Undriven
THC1_SPI2_IO[0:3]	Primary	Undriven	Undriven	Undriven
THC1_SPI2_RST#	Primary	Undriven	Undriven	Undriven
THC1_SPI2_INT#	Primary	Undriven	Undriven	Undriven
Note: 1. During reset refers to when PCH_RSMRST# is asserted.				

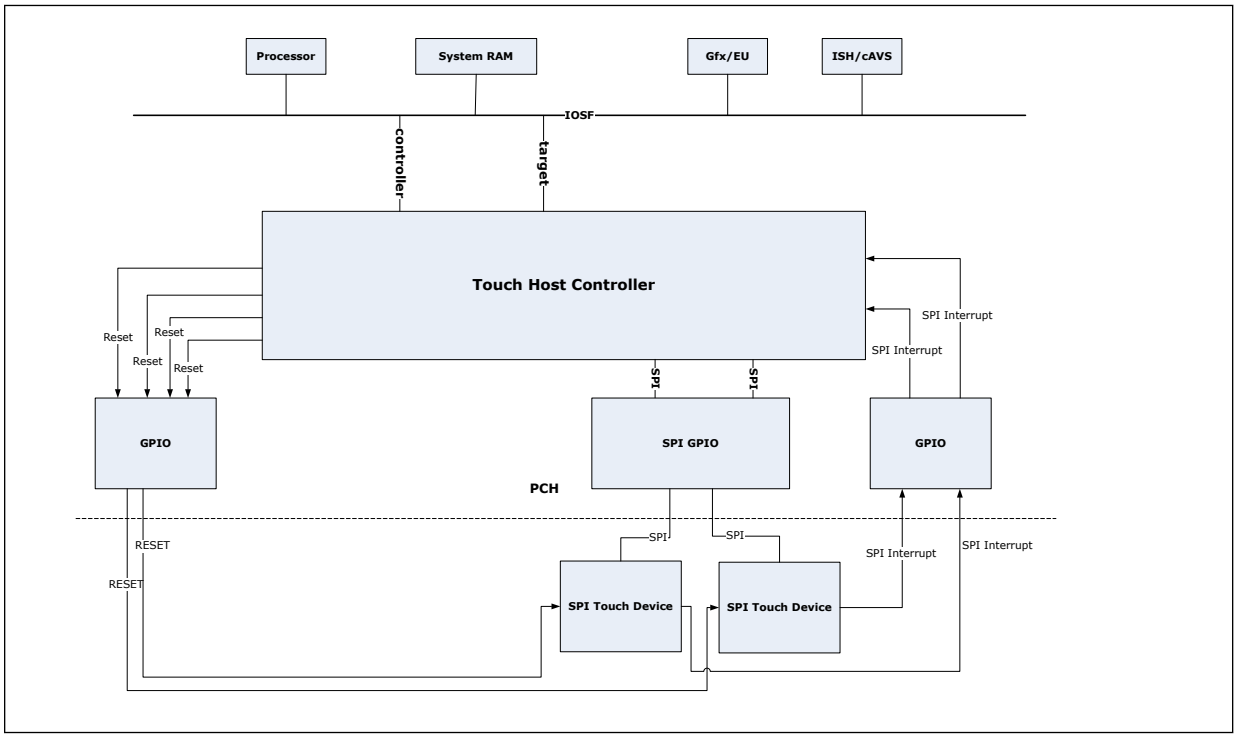
24.4 Functional Description

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 (For example, graphics EUs or host Processor), or host driver respectively.

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

- THC Controller
  - Touch Host controller bridges the PCH bus and SPI
  - The THC Controller has the following interfaces
- SPI IO
  - 1.8 V SPI IOs.
  - Provides SPI interface to the THC core.
  - Maximum Frequency supported 41.67 MHz.

Figure 18. THC Block Diagram



25.0 Universal Serial Bus (USB)

The Intel® 800 Series Chipset implement an xHCI USB 3.2 controller which provides support for up to 14 USB 2.0 signal pairs and 10 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 include USB 3.2 and USB 2.0 signaling.

Table 74. Acronyms

Acronyms	Description
xHCI	eXtensible Host Controller Interface

Table 75. References

Specification	Location
USB4* Specification	www.usb.org
USB 3.2 Specification	
USB 2.0 Specification	

25.1 Signal Description

Signal Name	Type	Description
USB32_1_RXN **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.
USB32_1_TXN **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.
USB32_2_RXN **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.
USB32_2_TXN **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.
USB32_3_RXN **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.

continued...

Signal Name	Type	Description
USB32_3_TXN **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.
USB32_4_RXN **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.
USB32_4_TXN **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.
USB32_5_RXN/ **USB32_5_RXP**	I	**USB 3.2 Differential Receive Pair 5**: These are USB 3.2-based high-speed differential signals for Port 5. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB32_5_TXN/ **USB32_5_TXP**	O	**USB 3.2 Differential Transmit Pair 5**: These are USB 3.2-based high-speed differential signals for Port 5. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB32_6_RXN/ **USB32_6_RXP**	I	**USB 3.2 Differential Receive Pair 6**: These are USB 3.2-based high-speed differential signals for Port 6. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB32_6_TXN/ **USB32_6_TXP**	O	**USB 3.2 Differential Transmit Pair 6**: These are USB 3.2-based high-speed differential signals for Port 6 . The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB32_7_RXN **USB32_7_RXP**	I	**USB 3.2 Differential Receive Pair 7**: These are USB 3.2-based high-speed differential signals for Port 7. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB32_7_TXN **USB32_7_TXP**	O	**USB 3.2 Differential Transmit Pair 7**: These are USB 3.2-based high-speed differential signals for Port 7. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB32_8_RXN **USB32_8_RXP**	I	**USB 3.2 Differential Receive Pair 8**: These are USB 3.2-based high-speed differential signals for Port 8. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB32_8_TXN **USB32_8_TXP**	O	**USB 3.2 Differential Transmit Pair 8**: These are USB 3.2-based high-speed differential signals for Port 8. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB32_9_RXN **USB32_9_RXP**	I	**USB 3.2 Differential Receive Pair 9**: These are USB 3.2-based high-speed differential signals for Port 9. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB32_9_TXN **USB32_9_TXP**	O	**USB 3.2 Differential Transmit Pair 9**: These are USB 3.2-based high-speed differential signals for Port 9. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.
USB32_10_RXN **USB32_10_RXP**	I	**USB 3.2 Differential Receive Pair 10**: These are USB 3.2-based high-speed differential signals for Port 10. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.

continued...

Signal Name	Type	Description
USB32_10_TXN **USB32_10_TXP**	O	**USB 3.2 Differential Transmit Pair 10**: These are USB 3.2-based high-speed differential signals for Port 10. 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.
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.
USB2P_11 **USB2N_11**	I/O	**USB 2.0 Port 11 Transmit/Receive Differential Pair 11:** 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_12 **USB2N_12**	I/O	**USB 2.0 Port 12 Transmit/Receive Differential Pair 12:** 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_13 **USB2N_13**	I/O	**USB 2.0 Port 13 Transmit/Receive Differential Pair 13:** 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_14 **USB2N_14**	I/O	**USB 2.0 Port 14 Transmit/Receive Differential Pair 14:** 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_H05/**USB_OC0#**/USB-C_GPP_H05	I	**Overcurrent Indicators**: This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred.

continued...

Signal Name	Type	Description
		When configured as OC# pin, a 10 kohm 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_H06/**USB_OC1#**/USB-C_GPP_H06	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 kohm 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_H07/**USB_OC2#**/USB-C_GPP_H07	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 kohm 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_H08/**USB_OC3#**/USB-C_GPP_H08	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 kohm 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_C16/**USB_OC4#**/I2C4_SDA/USB-C_GPP_C16	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 kohm 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_C17/**USB_OC5#**/I2C4_SCL/USB-C_GPP_C17	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 kohm 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.

continued...

Signal Name	Type	Description
GPP_C18/**USB_OC6#**/I2C5_SDA/USB-C_GPP_C18	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 kohm 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_C19/**USB_OC7#**/I2C5_SCL/USB-C_GPP_C19	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 kohm 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_ID1	I	ID detect for USB Device mode. *Note:* This HW signal is not used for dual role mode selection. The switching of USB port role is done through message from EC/PD over USB-C_SML. This signal should be connected to ground.
USB_ID2	I	ID detect for USB Device mode. *Note:* This HW signal is not used for dual role mode selection. The switching of USB port role is done through message from EC/PD over USB-C_SML. This signal should be connected to ground.
USB2_1_RCOMP	A	USB Resistor Bias, analog connection points for an external resistor 200 ohm ± 1% connected to GND.
USB2_2_RCOMP	A	USB Resistor Bias, analog connection points for an external resistor 200 ohm ± 1% connected to GND.
USB32_RCOMP	A	USB Resistor Bias, analog connection points for an external resistor 200 ohm ± 1% connected to GND.

25.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
USB2P_[14:1]	Internal Pull-down	14.25–24.8 kohm	1
USB2N_[14:1]	Internal Pull-down	14.25–24.8 kohm	1
Note: 1. Series resistors (45 ohm ±10%)			

25.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[2]	Immediately after Reset[2]	S3/S4/S5	Deep Sx
USB32_[10:1]_RXN	Primary	Internal Pull-Down	Internal Pull-Down	Internal Pull-Down	OFF

continued...

Signal Name	Power Plane	During Reset[2]	Immediately after Reset[2]	S3/S4/S5	Deep Sx
USB32_[10:1]_RXP					
USB32_[10:1]_TXN **USB32_[10:1]_TXP**	Primary	Internal Pull-Down	Internal Pull-Down	Internal Pull-Down	OFF
USB2N_[14:1]	DSW	Internal Pull-Down	Internal Pull-Down	Internal Pull-Down	Internal Pull-Down
USB2P_[14:1]	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_OC4#	Primary	Undriven	Undriven	Undriven	OFF
USB_OC5#	Primary	Undriven	Undriven	Undriven	OFF
USB_OC6#	Primary	Undriven	Undriven	Undriven	OFF
USB_OC7#	Primary	Undriven	Undriven	Undriven	OFF
USB_ID[2:1][1]	Primary	Internal Pull-Up	Undriven/Internal Pull-Up	Undriven/Internal Pull-Up	OFF
USB2_[2:1]_RCOMP	Primary	Undriven	Undriven	Undriven	OFF
USB32_RCOMP	Primary	Undriven	Undriven	Undriven	OFF

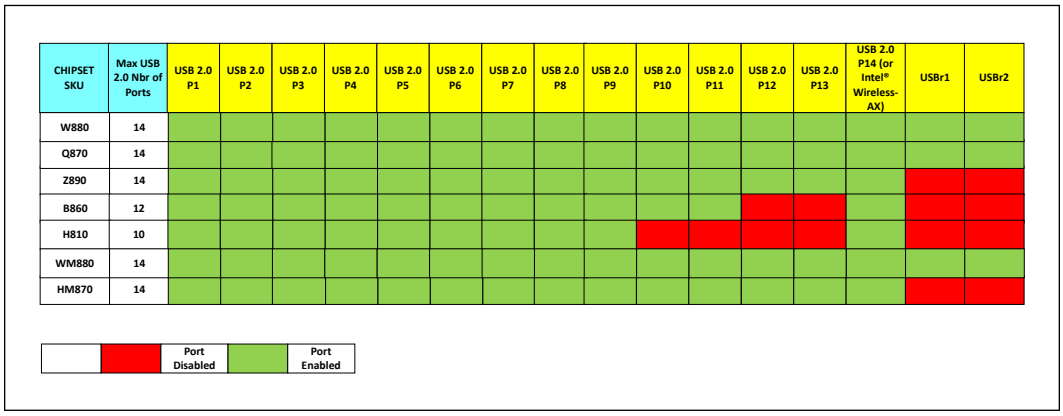
Notes: 1. The USB_ID pin is pulled up internally.
2. Reset reference for primary well pin is PCH_RSMRST# and DSW well pin is DSW_PWROK.

25.4 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:

- Intel® 800 Series Chipset
  - — USB 2.0 port 14 will be enabled on all platforms

Figure 19. Supported USB 2.0 Ports on Intel® 800 Series Chipset



25.5 Functional Description

25.5.1 eXtensible Host Controller Interface (xHCI) Controller

The eXtensible Host Controller Interface (xHCI) allows data transfer speed up to 20 Gb/s for USB 3.2 Gen 2x2 ports, 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 20 Gbps, 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.

25.5.2 USB Dual Role Support - eXtensible Device Controller Interface (xDCI) Controller

The USB subsystem also supports Dual Role Capability. The xHCI is paired with a stand-alone 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 supports SuperSpeed USB 5 Gbps, and High-Speed (HS). The device controllers is instantiated as a separate PCI function. The USB implementation is compliant to the Device specification and supports host/device only through the integrated 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 subsequent connection will not be established.

26.0 General Purpose Input and Output

The 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 either the PCH Primary well or Deep Sleep well.

The high level features of GPIO:

- Per-group configurable 3.3 V or 1.8 V voltage.
- Integrated pull-up / pull-down.
- Configurable as GPIO input, GPIO output, or native function signal.
- Configurable GPIO pad ownership by host, CSME, 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).

Table 76. 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

26.1 Functional Description

26.1.1 Configurable GPIO Voltage

Except for GPP_J, GPP_S, and GPD groups, all other GPIO groups support per-group configurable voltage as called out in the below table. The voltage of each GPIO group is selected by both connecting the corresponding GPIO power supply pin and setting the corresponding Group Voltage Select soft strap to the desired voltage (i.e. 3.3 V or 1.8 V).

Signal Coverage	Voltage Rail	Supported Voltages
All GPP_A signals	VCCPGPPA	1.8 V or 3.3 V
All GPP_B and GPP_C signals	VCCPGPPBC	1.8 V or 3.3 V
All GPP_D and GPP_R signals	VCCPGPPDR	1.8 V or 3.3 V
ALL GPP_E, GPP_F, GPP_H, GPP_K signals + SYS_RESET# and CL_RST#	VCCPGPPEFHK	1.8 V or 3.3 V
All GPP_I signals	VCCPGPPI	1.8 V or 3.3 V

NOTES

1. GPP_J and GPP_S groups support 1.8 V only.
2. GPD group supports 3.3 V only.

26.1.2 Interrupt / IRQ via GPIO Requirement

A GPIO, as an input, can be used to generate an interrupt / IRQ to the processor. In this case, it is required that the pulse width on the GPIO must be at least 100 us for the processor to recognize the interrupt.

26.1.3 Integrated Pull-ups and Pull-downs

All GPIOs have programmable internal pull-up/pull-down resistors. The internal pull-up/pull-down for each GPIO can be statically enabled/disabled by BIOS programming the corresponding TERM field in PAD_CFG_DW1 register (refer to Volume 2 for more details on the register).

NOTE

Although most of the GPIOs have the internal pull-up/pull-down disabled by default, some of them have the internal pull-up/pull-down enabled by default. Refer to the GPIO Implementation Summary document for detailed info on default termination.

26.1.4 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 GPIOs that can be routed to generate SMI or NMI:

- GPP_B14, GPP_B[21:20]
- GPP_C[23:22]
- GPP_D[04:00]
- GPP_E[08:00]
- GPP_I[03:00]

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

Certain GPIOs are capable of supporting Blink (BK) and Serial Blink (SBK). The BK and SBK are implemented as native functions muxed on selected GPIOs. To enable BK or SBK on a GPIO having the capability, BIOS needs to select the BK or SBK native function on the GPIO.

Blink, also referred to as Pulse Width Modulator (PWM), provides a programmable PWM wave. The Blink/PWM frequency and duty cycle are programmable through the PWM Control Register. Refer to Volume 2 for the related register.

Serial Blink, also referred to as Serial POST Codes, allows system software to serialize POST or other messages 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 the related registers.

26.1.6 GPIO Ownership

A 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 Intel® CSME or ISH, the host software no longer has access to the pin programming.

26.1.7 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 77. 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, HDAPROC_SDI, HDAPROC_SDO, I2S[2:0]_SCLK, I2S[2:0]_SFRM, I2S[2:0]_RXD, DMIC_DATA[1:0], SNDW[3:0]_DATA
Power Management	ACPRESENT, LAN_WAKE#
Thunderbolt™ / BSSB	BSSB_LS_RX, BSSB_LS_TX
I3C	I3C0_SDA, I3C0_SCL; BPKI3C_SDA, BPKI3C_SCL
ISH	ISH_I3C0_SDA, ISH_I3C0_SCL

26.2 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, **833778-001_GPIO.xlsx**.

27.0 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.

27.1 Signal Description

Signal Name	Type	Description
GPP_K00/**GSXDOUT**	O	GPIO Serial Expander Controller Data Out
GPP_K01/**GSXSLOAD**	O	GPIO Serial Expander Controller Serial Load
GPP_K02/**GSXDIN**	I	GPIO Serial Expander Controller Data In
GPP_K03/**GSXSRESET#**	O	GPIO Serial Expander Controller Serial Reset
GPP_K04/**GSXCLK**	O	GPIO Serial Expander Controller Clock

27.2 Integrated Pull-ups and Pull-downs

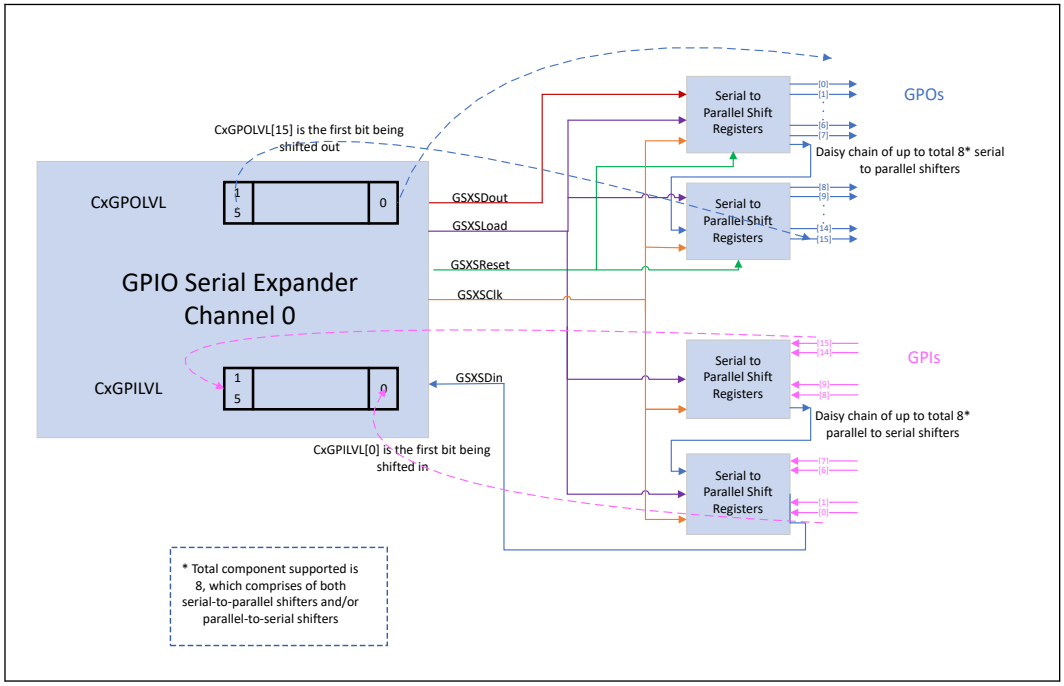
None

27.3 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 20. GSX Topology - Example



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.

28.0 Private Configuration Space 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 Private Configuration Space at individual endpoints (Target Port IDs) which are only accessible through the Sideband Interface. These Private Configuration Space Registers can be addressed via SBREG_BAR or through SBI Index Data pair programming.

Table 78. Private Configuration Space Register Target Port IDs

Device/Function Type	Target Port ID (hex)
DCI	B8h
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 3	6Bh
General Purpose I/O (GPIO) Community 4	6Ah
General Purpose I/O (GPIO) Community 5	69h
SATA	D9h
SMBus	C6h
eSPI / SPI	72h
xHCI	70h
USB 3.2 Gen 1x1 Dual Role (xDCI)	71h

29.0 Security Firmware Engines

29.1 Intel® Converged Security and Management Engine (Intel® CSME)

CSxE is a security engine which provides security firmware authentication and loading, secure boot, platform debug control, and manageability via Intel® Active Management Technology (Intel® AMT).

CSxE has a standalone small x86 processor, memory, crypto engine, and I/O's.

CSxE is isolated in a secured hardware and firmware environment from host processors.

29.2 Intel® Silicon Security Engine

A Security engine which is HW IP is based on CSxE HW IP and new FW IP design to be silicon Root of Trust providing secure FW loading, measurements and on-tile certification authority.

The firmware is based on a new design which focus on security, simplicity of architecture and isolated environment.

30.0 System Management

The 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:

- 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
- 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 79. Acronyms

Acronyms	Description
BMC	Baseboard Management Controller
EC	Embedded Controller
SPD	Serial Presence Detect
TCO	Total Cost of Ownership

30.1 Theory of Operation

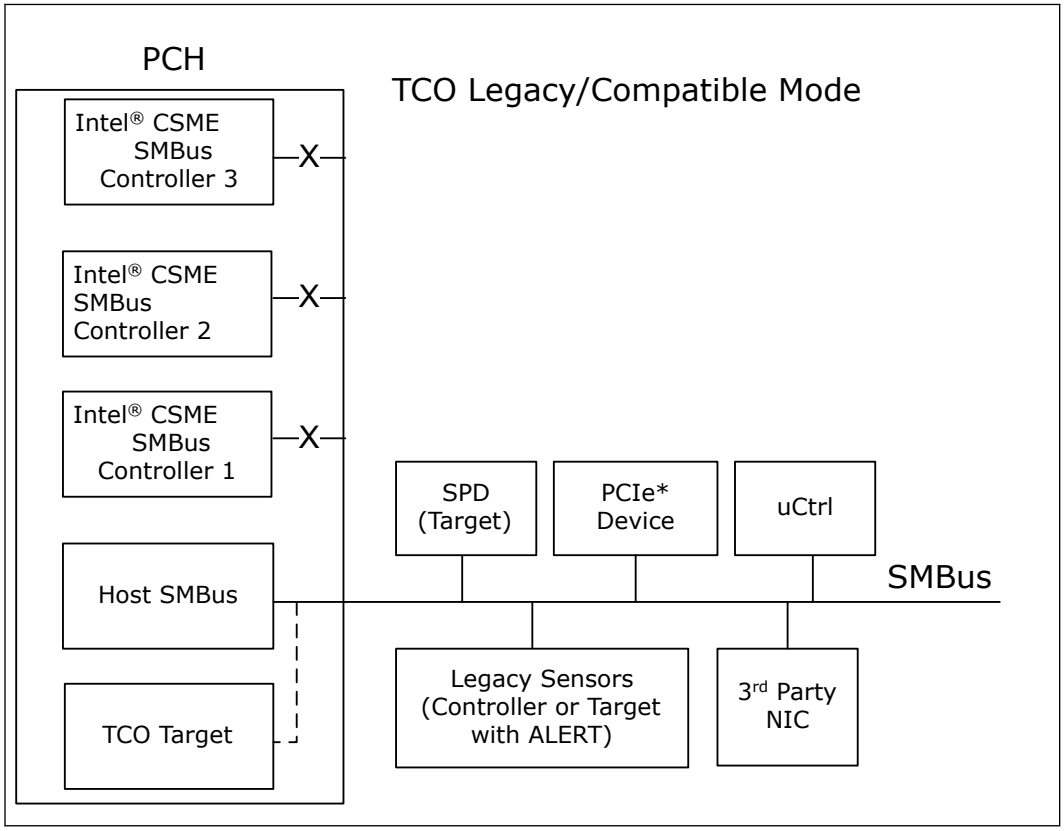
The System Management functions are designed to allow the system to diagnose failing subsystems. The intent of this logic is that some of the system management functionality can be provided without the aid of an external microcontroller.

30.1.1 TCO Modes

TCO Compatible Mode

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

Figure 21. 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. The table below includes a list of events that will report messages to the network management console.

Table 80. 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
SYSPWR_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.

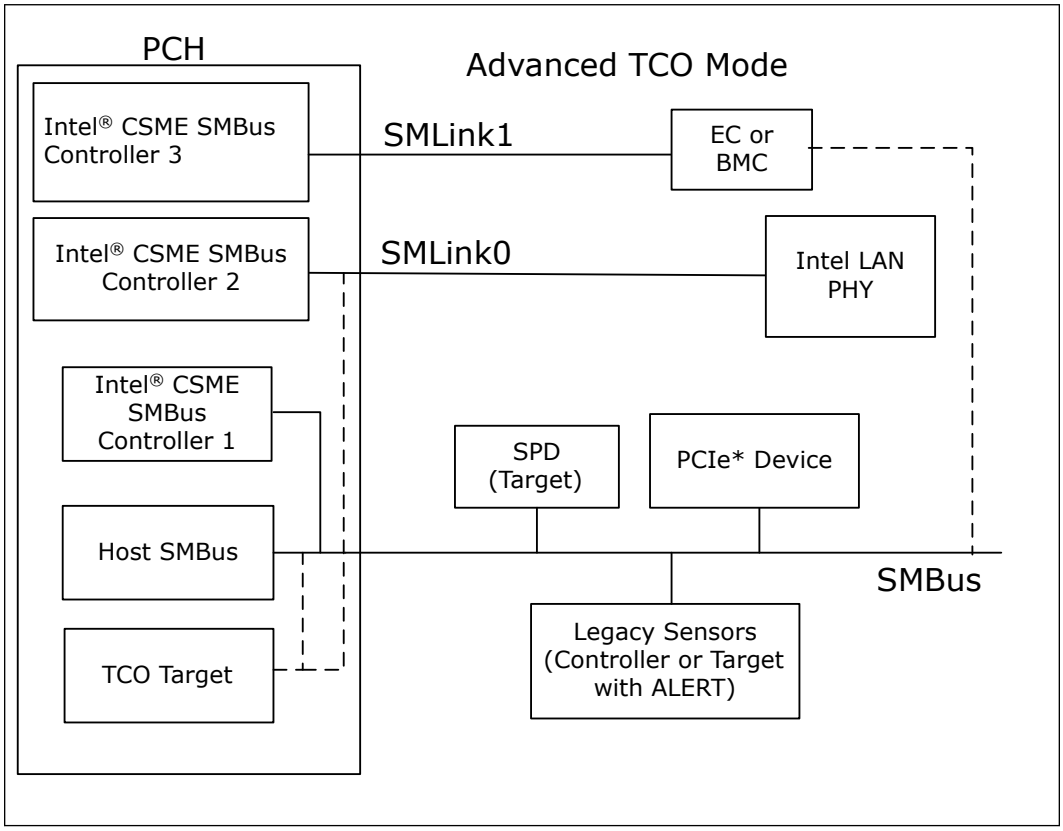
In this mode, the Intel® CSME SMBus controllers must be enabled by soft strap in the flash descriptor. Refer to figure below for more details.

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

SMLink0 is targeted 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. When the Fast Mode is enabled using a soft strap, the interface will be running at the frequency of up to 1 MHz depending on different factors such as board routing or bus loading.

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

Figure 22. Advanced TCO Mode



31.0 Testability and Monitoring

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

Table 81. Acronyms

Acronyms	Description
BSDL	Boundary Scan Description Language
DCI	Direct Connect Interface
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input/Output
I/OD	Input/Output Open Drain
Intel® TH	Intel® Trace Hub
JTAG	Joint Test Action Group

Table 82. References

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

31.1 Signal Description

Table 83. Testability Signals

Signal Name	Type	Description
PCH 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	IOD	**Test Mode Select (TMS):** The signal is decoded by the Test Access Port (TAP) controller to control test operations.
PCH_JTAG_TDI	IOD	**Test Data Input (TDI):** Serial test instructions and data are received by the test logic at TDI.
PCH_JTAG_TDO	IOD	**Test Data Output (TDO):** TDO is the serial output for test instructions and data from the test logic defined in this standard.
PCH_JTAG_TRST#	I/O	**Test Reset(TRST)**: Resets the Test Access Port (TAP) logic. This signal should be driven low during power-on Reset.
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.
PRDY#	IOD	**Probe Mode Ready:** PRDY# is a processor output used by debug tools to determine processor debug readiness.

continued...

Signal Name	Type	Description
PREQ#	IOD	**Probe Mode Request:** PREQ# is used by debug tools to request debug operation of the processor.
TRIGGER_IN	I	This pin is used for debug support.
TRIGGER_OUT	O	This pin is used for debug support.
GPP_J11/**DAM**	O	Debug Authentication Mode.
Boundary Scan Sideband Signals and Debug for I3C		
GPP_J10/**BPKI3C_SCL**/**BSSB_LS_TX**	I/O	**BPKI3C_SCL:**Debug for I3C Clock **BSSB_LS_TX:** Boundary Scan Sideband Low Speed Transmit for debug purposes
GPP_J09/**BPKI3C_SDA**/**BSSB_LS_RX**	I/O	**BPKI3C_SDA:**Debug for I3C Data **BSSB_LS_RX:**Boundary Scan Sideband Low Speed Receive for debug purposes
Boot Halt Signal and Fuse Break		
GPP_J07/**BOOTHALT#**	IOD	**Boot Halt:** This signal is used for platform boot halt.

Table 84. Testability Signals

31.2 I/O Signal Planes and States

Table 85. Power Planes and States for Testability Signals

Signal Name	Power Plane[2]	Resistors [2 3]	During Reset[1]	Immediately after Reset[1]	S4/S5
PCH JTAG signals					
PCH_JTAG_TCK	Primary	Internal Pull-Down	Driven Low	Driven Low	Driven Low
PCH_JTAG_TMS	Primary	Internal Pull-Up	Driven High	Driven High	Driven High
PCH_JTAG_TDI	Primary	Internal Pull-Up	Driven High	Driven High	Driven High
PCH_JTAG_TDO	Primary	External Pull-Up	Undriven	Undriven	Undriven
PCH_JTAG_TRST#	Primary	Internal Pull-Down	Driven Low	Driven Low	Driven Low
DBG_PMODE	Primary	Internal Pull-Up	Driven High	Driven High	Driven High
Note: 1. Reset reference for primary well pins is PCH_RSMRST#.					

32.0 Miscellaneous Signals

32.1 Signal Description

Table 86. Signal Descriptions

Signal Name	Type	Description
GPP_E13/SBK0/**BK0**	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_E14/SBK1/**BK1**	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_E15/SBK2/**BK2**	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_E16/SBK3/**BK3**	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_F19/ **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 shut down 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_E00/**SATAXPCIE0**/ SATAGP0	I	**SATA port 0 or PCIe port mux select** : This pin can be used as a SATA/PCIE mux selector with the polarity (SATA or PCIe) controlled by soft-strap. SATAXPCIE native function mode is also determined by soft strap.
GPP_E01/**SATAXPCIE**1/ SATAGP1	I	**SATA port 1 or PCIe port mux select** : This pin can be used as a SATA/PCIE mux selector with the polarity (SATA or PCIe) controlled by soft-strap. SATAXPCIE native function mode is also determined by soft strap.
GPP_E02/**SATAXPCIE2**/ SATAGP2	I	**SATA port 2 or PCIe port mux select** : This pin can be used as a SATA/PCIE mux selector with the polarity (SATA or PCIe) controlled by soft-strap. SATAXPCIE native function mode is also determined by soft strap.
GPP_F00/**SATAXPCIE3**/ SATAGP3	I	**SATA port 3 or PCIe port mux select** : This pin can be used as a SATA/PCIE mux selector with the polarity (SATA or PCIe) controlled by soft-strap. SATAXPCIE native function mode is also determined by soft strap.
GPP_F01/**SATAXPCIE4**/ SATAGP4	I	**SATA port 4 or PCIe port mux select** : This pin can be used as a SATA/PCIE mux selector with the polarity (SATA or PCIe) controlled by soft-strap. SATAXPCIE native function mode is also determined by soft strap.
GPP_F02/**SATAXPCIE5**/ SATAGP5	I	**SATA port 5 or PCIe port mux select** : This pin can be used as a SATA/PCIE mux selector with the polarity (SATA or PCIe) controlled by soft-strap. SATAXPCIE native function mode is also determined by soft strap.

continued...

Signal Name	Type	Description
GPP_F03/**SATAXPCIE6**/ SATAGP6	I	**SATA port 6 or PCIe port mux select** : This pin can be used as a SATA/PCIE mux selector with the polarity (SATA or PCIe) controlled by soft-strap. SATAXPCIE native function mode is also determined by soft strap.
GPP_F04/**SATAXPCIE7**/ SATAGP7	I	**SATA port 7 or PCIe port mux select** : This pin can be used as a SATA/PCIE mux selector with the polarity (SATA or PCIe) controlled by soft-strap. SATAXPCIE native function mode is also determined by soft strap.
GPP_E13/**SBK0**/BK0	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_E14/**SBK1**/BK1	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_E15/**SBK2**/BK2	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.
GPP_E16/**SBK3**/BK3	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.

32.2 Integrated Pull-Ups and Pull-Downs

Table 87. Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value
SATAXPCIE0	Pull-down	20 kohm ± 30%
SATAXPCIE1	Pull-down	20 kohm ± 30%
SATAXPCIE2	Pull-down	20 kohm ± 30%
SATAXPCIE3	Pull-down	20 kohm ± 30%
SATAXPCIE4	Pull-down	20 kohm ± 30%
SATAXPCIE5	Pull-down	20 kohm ± 30%
SATAXPCIE6	Pull-down	20 kohm ± 30%
SATAXPCIE7	Pull-down	20 kohm ± 30%

32.3 Ground and Reserved Signals

The following are the general types of reserved (RSVD) signals and connection guidelines:

- RSVD – these signals should not be connected
- RSVD_TP – these signals should be routed to a test point

Arbitrary connection of these signals to VCC, VSS, or to any other signal (including each other) may result in component malfunction or incompatibility with future processors. Refer to the table below.

For reliable operation, always connect unused inputs or bi-directional signals to an appropriate signal level. Unused active high inputs should be connected through a resistor to ground (VSS). Unused outputs may be left unconnected however, this may interfere with some Test Access Port (TAP) functions, complicate debug probing and prevent boundary scan testing. A resistor should be used when tying bi-directional signals to power or ground. When tying any signal to power or ground the resistor can also be used for system testability. Resistor values should be within ±20% of the impedance of the baseboard trace, unless otherwise noted.

Table 88. GND and RSVD Signals

Signal Name	Description
VSS	**Ground:** PCH ground node
RSVD	**Reserved**: All signals that are RSVD should not be connected on the board.
RSVD_TP	**Test Point:** Intel recommends to route each RSVD_TP to an accessible test point. Intel may require these test points for platform specific debug. Leaving these test points inaccessible could delay debug by Intel.

33.0 Processor Sideband Signals

The sideband signals are used for the communication between the interfaces within the processor.

Table 89. Acronyms

Acronyms	Description
PECI	Platform Environmental Control Interface

33.1 Signal Description

Signal Name	Type	Description
GPP_E03/**PROC_GP0**	I	Thermal management signal
GPP_E07/**PROC_GP1**	I	Thermal management signal
GPP_B03/**PROC_GP2**	I	Thermal management signal
GPP_B04/**PROC_GP3**	I	Thermal management signal
GPP_J15/**AUX_PWRGD**	O	Signal from PCH to indicate that VNN Power is good. This signal only applicable to Workstation SKU
PROCPWRGD	O	Signal from PCH to indicate its primary power is good.. This signal only applicable to Workstation SKU

33.2 I/O Signal Planes and States

Signal Name	Power Plane	During Reset[1]	Immediately after Reset [1]	S4/S5
PROC_GP[3:0]	Primary	Undriven	Undriven	Undriven
Note: 1. Reset reference for primary well pins is PCH_RSMRST#.				

34.0 Power Delivery

34.1 Power and Ground Signals

This section describes the PCH power rails.

Table 90. Power Rail Descriptions

Name	Description
VCCPRIM_CORE_0P82	**Primary Core Voltage:** 0.82 V
VCCDUSB_0P82	**USB digital logic voltage:** 0.82 V
VCCHSIO_0P82	**HSIO Supply Voltage**: 0.82 V
VCCAHSIOPLL_1P8	**Analog supply for HSIO PLLs:** 1.8 V
VCCA_XTAL_PLL_1P8	**Analog supply for XTAL circuit:** 1.8 V
VCCA_XTAL_PLL_0P82	**Analog supply for XTAL circuit:** 0.82 V
VCCDSW_3P3	**Deep Sx Well Voltage:** 3.3 V
VCCPRIM_1P8	**Primary Well**: 1.8 V
VCCPRIM_3P3	**Primary Well**: 3.3 V
VCCSPI	**SPI Voltage:** 3.3 V or 1.8 V.
VCCPRIM_JTAGPROC	This rail should be connected to the Processor VCCPRIM_IO_OUT_PCH rail
DCPRTC	This rail is generated internally and needs to be routed out to the motherboard for decoupling purpose.
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.
DCPCNVILDO1	This rail is generated internally and needs to be routed out to the motherboard for decoupling purpose.
DCPCNVILDO2	This rail is generated internally and needs to be routed out to the motherboard for decoupling purpose.
VCCPGPPA	Power rail for GPP_A group.
VCCPGPPBC	Power rail for GPP_B and GPP_C group.
VCCPGPPDR	Power rail for GPP_D and GPP_R group.

continued...

Name	Description
VCCPGPPEFHK	Power rail for GPP_E, GPP_F,GPP_H and GPP_K group.
VCCPGPPI	Power rail for GPP_I group.
VSS	Ground

35.0 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.

Table 91. Acronyms

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

Table 92. References

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

35.1 System Power States, Advanced Configuration and Power Interface (ACPI)

This section describes System Power States and ACPI states supported by the PCH.

Table 93. General System Power States

State	Description
G0/S0/C0	**Full On**: Processor operating. Individual devices may be shut to save power. The different Processor operating levels are defined by Cx states.
GO/S0/Cx	**Cx state**: Processor manages C-states by itself and can be in low power state
G1/S3	**Suspend-To-RAM (STR)**: The system context is maintained in system DRAM, but power is shut to non-critical circuits. Memory is retained, and refreshes continue. All external clocks are shut off; RTC clock and internal ring oscillator clocks are still toggling. In S3, SLP_S3 signal stays asserted, SLP_S4 and SLP_S5 are inactive until a wake occurs.
G1/S4	**Suspend-To-Disk (STD)**: The context of the system is maintained on the disk. All power is then shut to the system except to the logic required to resume. Externally appears same as S5 but may have different wake events. In S4, SLP_S3 and SLP_S4 both stay asserted and SLP_S5 is inactive until a wake occurs. In S4, SLP_S4 would stay asserted and SLP_S5 is inactive until a wake occurs
G2/S5	**Soft Off**: System context not maintained. All power is shut except for the logic required to restart. A full boot is required when waking. Here, SLP_S3, SLP_S4, and SLP_S5 are all active until a wake occurs.

continued...

State	Description
	Here, SLP_S4, and SLP_S5 are all active until a wake occurs.
G3	**Mechanical OFF**: System context not maintained. All power shut except for the RTC. No "Wake" events are possible because the system does not have any power. This state occurs if the user removes the batteries, turns off a mechanical switch, or if the system power supply is at a level that is insufficient to power the "waking" logic. When system power returns the transition will depend on the state just prior to the entry to G3.

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 94. State Transition Rules for the PCH

Present State	Transition Trigger	Next State
G0/S0/C0	• SLP_EN bit set • Power Button Override[3] • Mechanical Off/Power Failure	• G0/S0/Cx • G1/S4, or G2/S5 state • G2/S5 • G3
G0/S0/Cx	• Power Button Override[3] • Mechanical Off/Power Failure	• G0/S0/C0 • S5 • G3
G1/S3	• Any Enabled Wake Event • Power Button Override[3] • Mechanical Off/Power Failure	• G0/S0/C0[2] • G2/S5 • G3
G1/S4	• Any Enabled Wake Event • Power Button Override[3] • Mechanical Off/Power Failure	• G0/S0/C0[2] • G2/S5 • G3
G2/S5	• Any Enabled Wake Event • Mechanical Off/Power Failure	• G0/S0/C0[2] • G3
G2	• Any Enabled Wake Event • Mechanical Off/Power Failure • Power Button Override	• G0/S0/C0[2] • G1/S4 or G2/S5 • G3 • G2/S5
G3	• Power Returns	• S0/C0 (reboot) or G2/S5[4] (stay off until power button pressed or other wake event)[1,2]

Notes: 1. Some wake events can be preserved through power failure.
2. Transitions from the 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.

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 95. 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)	SLP_S3# signal	When SLP_S3# goes active, power can be shut off to any circuit not required to wake the system The PCH 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_PRIM#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_PRIM#	This signal is asserted when the Primary rails can be externally shut off for enhanced power saving.
VCCIO and VCCSTG	CPU_C10_GATE#	This signal is asserted when the processor enters C10 and can handle VCCIO, VCCSTG and VCCPLL_OC being lowered to 0 V.
DEVICE[n]	GPIO	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.

35.2 Legacy Power Management Support

Read Only Registers with Write Paths in ALT Access Mode

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

Table 96. Register Write Accesses in ALT Access Mode

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

35.2.1 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 processor implements an ALT access mode.

If the ALT access mode is entered and exited after reading the registers of the 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 processor 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 97. Write Only Registers with Read Paths in ALT Access Mode

Restore Data			
I/O Addr	# of Rds	Access	Data
70h	1		Bit 7 = Read value is ‘0’. Bits [6:0] = RTC Address

35.2.2 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 processor 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 processor 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.

35.3 Functional Description

This section provides information on the following topics:

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

35.3.1 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
- Intel® CSME Power Management Support
  - — Wake events from the Intel® CSME (enabled from all S-States including Catastrophic S5 conditions)

35.3.2 Power Saving Features

Power Management Substates

S0i2.0, S0i2.1, S0i2.2 are the supported Power management substates. If a platform wants to enable/disable the S0ix states, BIOS can do by modifying the LPM_EN register.

35.3.3 SMI#/SCI Generation

Upon any enabled SMI event taking place while the End of SMI (EOS) bit is set, the processor will clear the EOS bit and assert SMI, 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 processor 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 processor 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 98. 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 [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	

continued...

Cause	SCI	SMI	Additional Enables[1]	Where Reported
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
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 [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 Device 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

continued...

Cause	SCI	SMI	Additional Enables[1]	Where Reported
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 Vol2	RTC_UIP_SMI_STS
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 processor must have SMI fully enabled when the processor 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 section for specific GPIOs enabled for SCIs and/or SMIs
9. Secondary eSPI 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 processor will set the PCI_EXP_STS bit. If the PCI_EXP_EN bit is also set, the processor 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.

35.3.4 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 controller 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.
- Shutdown by integrated manageability functions (ASF/Intel® CSME).
- Internal watchdog timer timeout events.

Table 99. Sleep Types

Sleep Type	Comment
S4	The PCH asserts 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_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 100. Causes of Wake Events

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

continued...

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

Notes:
1. If BATLOW# signal is low, PCH will not attempt to wake from S4/S5 , 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 .
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 controller 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.

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 PCH_RSMRST#.

Table 101. 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
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 PRIMPWRDNACK 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 102. 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
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 PRIMWARN#/PRIMACK# handshake to ensure the platform is ready to enter Deep Sx. The PCH asserts PRIMWARN# as notification that it is about to enter Deep Sx. Before the PCH proceeds and asserts SLP_PRIM#, the PCH waits for PRIMACK# 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_PRIM#.

Table 103. 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.

35.3.5 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 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 and G3/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 104. 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]
G3	PWRBTN# pressed	None	No effect since no power Not latched nor detected *Notes:* 1. During G3 exit, PWRBTN# pin must be kept de-asserted for a minimum time of 500 us after the PCH_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 *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 (replaced PME_B0 bit) that will be set by the PCH when any internal device with PCI Power Management capabilities on bus 0 asserts the equivalent of the PME# signal. This is separate from the external PME# signal and can cause the same effect.

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.

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 PCH_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 (PCH_RSMRST# is asserted), the EC must not allow PCH_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, SX_EXIT_HOLDOFF# functionality is not available, and assertion of the signal will not impact Sx exit flows.

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

The SLP_S4# output signal is used to remove power to additional subsystems that are powered during SLP_S3#, (moved from SLP_S5# description to SLP_S4#) 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. (moved to SLP_S4# description --- as well as power to the system memory, since the context of the system is saved on the disk. Cutting power to the memory may be done using the power supply, or by external FETs on the motherboard.)

SLP_A# output signal can be used to cut power to the Intel® 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 PCH_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'.

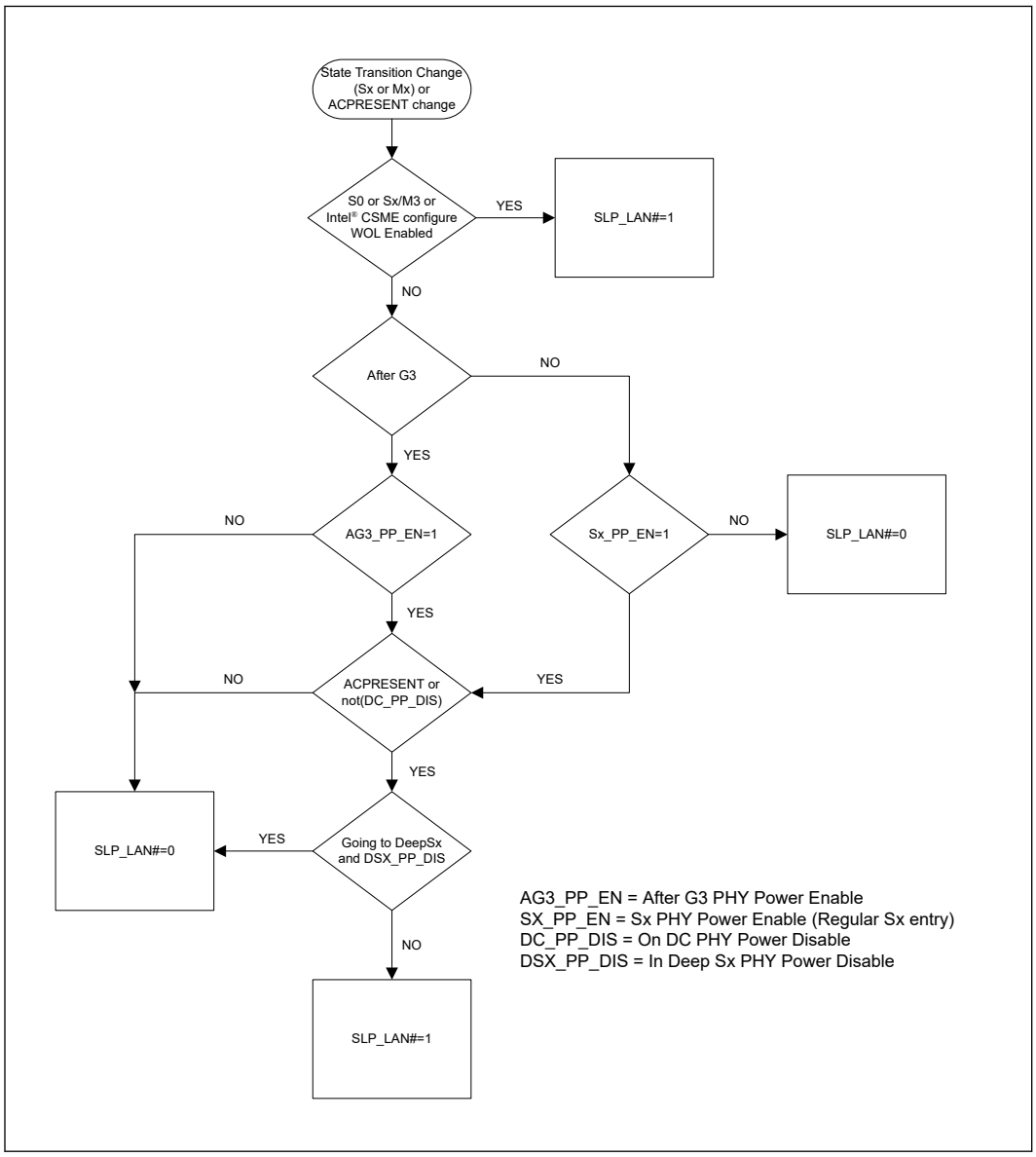
- If the LAN PHY is required by Intel® CSME in Sx/M-Off or Deep Sx, Intel® CSME must configure SLP_LAN#='1' irrespective of the power source and the destination power state. Intel® CSME must be powered at least once after G3 to configure this.
- If the LAN PHY is required after a G3 transition, the host 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 23. 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.

PRIMPWRDNACK Steady State Pin Behavior

Below table summarizes PRIMPWRDNACK pin behavior.

Table 105. PRIMPWRDNACK/PRIMWARN#/GPP_A02 Pin Behavior

Pin	GPP_A02 Input/ Output (Determine by GP_IO_SEL bit)	Pin Value in S0	Pin Value in Sx/M-Off	Pin Value in Sx/M3
PRIMPWRDNACK	Native	0	Depends on Intel® CSME power package and power source (Note 1)	0
PRIMWARN#	Native	1	1	1
GPP_A02	IN	High-Z	High-Z	High-Z
	OUT	Depends on GPP_A02 output data value	Depends on GPP_A02 output data value	Depends on GPP_A02 output data value

Table 106. PRIMPWRDNACK 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 107 on page 207	

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.

35.3.7 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 processor times out 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 107. 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 Secondary Message received for 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)
SMBus Secondary Message received for Reset without Power-Cycle	Yes	No	No[4]	
SMBus Secondary Message received for unconditional Power Down	No	No	No	Yes
TCO Watchdog Timer reaches zero two times	Yes	No	No[4]	
Power Failure: PLT_PWROK signal goes inactive in S0	No	No	Yes	
SYS_PWROK Failure: SYS_PWROK signal goes inactive in S0	No	No	Yes	
Processor 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]	
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[7]	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 6)	No	No	Yes	
PLT_PWROK Stuck Low	No	No	Yes	

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)
Power Management Watchdog Timer	No	No	No[7]	Yes
Intel® Converged Security and Management Engine Hardware Uncorrectable Error	No	No	No[7]	Yes

Notes:
1. The PCH drops this type of reset request if received while the system is in S4/S5.
2. The 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. PLTRST# Entry Timeout is automatically initiated if the hardware detects that the PLTRST# sequence has not been completed within 4 seconds of being started.
7. 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.

35.3.8 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 processor 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 processor 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.

35.4 Power and Internal Signals

35.4.1 Signal Description

Signal Name	Type	Description
GPD01/**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.
GPD00/**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:* An external pull-up 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. For non-Deep Sx implementation DSW_PWROK is tied to PCH_RSMRST#. DSW_PWROK goes high with PCH_RSMRST# after PRIM rail complete ramp For Deep Sx Implementation DSW_PWROK and PCH_RSMRST# are separate signals and should not be tied
GPD02/**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, is an indication to the PCH that all of its core power rails have been stable. The platform may drive asynchronously. When is de-asserted, the PCH asserts PLTRST#. *Notes:* • must not glitch, even if PCH_RSMRST# is low • An external pull-down resistor is required.
GPP_B13/**PLTRST#**	O	**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_B18/**PMCALERT#**	I/OD	**PMC Alert Pin**: Supports USB-C* PD controller architecture. *Note:* An External Pull up Resistor is required regardless of whether the function is used or not.
GPD03/**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.
PCH_RSMRST#	I	**Primary Well Reset**: This signal is used for resetting the primary power plane logic. This signal must be asserted for at least 10ms before de-asserting. *Note:* An external pull down resistor is required.
GPD06/**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

continued...

Signal Name	Type	Description
		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.
SLP_LAN#	IO	**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.
GPD09/**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.
GPD04/**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.
GPD05/**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. *Note:* • This pin must be used to control the DRAM power in order to use the PCH DRAM power-cycling feature.
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.
SLP_PRIM#	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-DSx, it should be No Connect. *Notes:* • This is in the DSW power well • An external pull-down resistor is required.
GPP_A03/ESPI_IO3/**PRIMACK#**	I	**PRIMACK#**: If Deep Sx is supported, the EC/motherboard controlling logic must change PRIMACK# to match PRIMWARN# once the EC/motherboard controlling logic has completed the preparations discussed in the description for the PRIMWARN# pin. *Note:* PRIMACK# is only required to change in response to PRIMWARN# if Deep Sx is supported by the platform.
GPD08/**SUSCLK**	O	**Suspend Clock**: This clock is a digitally buffered version of the RTC clock.
GPP_A02/ESPI_IO2/**PRIMWARN#**/ PRIMPWRDNACK	O	**PRIMWARN#**: This pin asserts low when the PCH is planning to enter the Deep Sx power state and remove Primary power (using SLP_PRIM#). 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. PRIMACK# must be driven to match PRIMWARN# once the above preparation is complete. PRIMACK# should be asserted within a minimal amount of time from PRIMWARN# assertion as no wake events are supported if PRIMWARN# is asserted but PRIMACK# is not asserted. Platforms supporting Deep Sx, but not wishing to participate in the handshake during wake and Deep Sx entry may tie PRIMACK# to PRIMWARN#. This pin is multiplexed with PRIMPWRDNACK since it is not needed in Deep Sx supported platforms.
GPP_A02/ESPI_IO2/PRIMWARN#/ **PRIMPWRDNACK**	O	**PRIMPWRDNACK**: 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.

continued...

Signal Name	Type	Description
GPP_B05/**SX_EXIT_HOLDOFF#**/ ISH_GP6	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_RESET#	I	**System Reset**: This pin forces an internal reset after being de-bounced. *Note:* An external pull-up resistor is required.
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 cannot be used to wake up the system during S0ix states. • An external pull-up resistor is required.
GPP_J16/**GLB_RST_WARN#**	I	An early indication that the Processor has detected a condition that will cause it to signal the system to perform a power cycle *Note:* An external pull-up resistor is required.
RESET_SYNC#	IOD	Bidirectional signal used to synchronize reset events between the processor and PCH. Sync reset exits with the PCH in following two steps: 1. The processor floats the RESET_SYNC# pin to allow it to be pulled high by the platform. 2. The processor waits until it sees the RESET_SYNC# pin go high before proceeding to reset exit. The PCH will stop driving the pin low when the eSPI controller in the PCH is out of reset and ready to receive traffic over the eSPI link. It is required that the PCH eSPI controller is ready before the Processor eSPI controller is released from reset.

35.4.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value
LAN_WAKE#	Pull-down	20 kohm +/- 30%
PWRBTN#	Pull-up	20 kohm +/- 30%
PRIMACK#	Pull-up	20 kohm+/- 30%
WAKE#	Pull-down	20 kohm +/- 30%

35.4.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset	Immediately after Reset	S4/S5	Deep Sx
ACPRESENT	DSW	Undriven /Driven Low4	Undriven	Undriven	Undriven/Internal Pull-down
BATLOW#	DSW	Undriven	Undriven	Undriven	Undriven
DSW_PWROK	RTC	Undriven	Undriven	Undriven	Undriven
LAN_WAKE#	DSW	Undriven	Undriven	Undriven	Undriven/Internal Pull-down
LANPHYPC	DSW	Driven Low	Driven Low / Driven High	Driven Low	Driven Low
PCH_PWROK	RTC	Undriven	Undriven	Undriven	Undriven

continued...

Signal Name	Power Plane	During Reset	Immediately after Reset	S4/S5	Deep Sx
PLTRST#	Primary	Driven Low	Driven High	Driven Low	OFF
PWRBTN#	DSW	Internal Pull-up	Internal Pull-up	Internal Pull-up	Internal Pull-up
PCH_RSMRST#	RTC	Undriven	Undriven	Undriven	Undriven
SLP_LAN#	DSW	Driven Low	Driven Low	Driven High/ Driven Low	Driven High/ Driven Low
SLP_S4#	DSW	Driven Low	Driven High	Driven Low	Driven Low
SLP_S5#	DSW	Driven Low	Driven High	Driven High/ Driven Low	Driven High/ Driven Low
SLP_PRIM#	DSW	Driven Low	Driven High	Driven High	Driven Low
SLP_WLAN#	DSW	Driven Low	Driven Low	Driven Low / Driven High	Driven Low / Driven High
PRIMACK#	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up	OFF
SUSCLK	DSW	Driven Low	Toggling	Toggling	Toggling
PRIMWARN# / PRIMPWRDNACK	Primary	Driven Low	Driven Low	Driven Low	OFF
SX_EXIT_HOLDOFF#	Primary	Undriven	Undriven	Undriven	OFF
SYS_RESET#	Primary	Undriven	Undriven	Undriven	OFF
WAKE#	DSW	Undriven	Undriven	Undriven	Undriven/Internal Pull-down
Note: Notes will be updated in the next revision.					

36.0 Thermal Sensor

The PCH incorporates on-die Digital Thermal Sensors for thermal management.

36.1 Modes of Operation

The Thermal sensors have two usages when enabled:

1. One use is to provide the temperature of the PCH in units of 1 ºC. There is a 8 bit field for the temperature, with a theoretical range from –128 ºC to +127 ºC. Practically the operational range for the system is between -40 ºC and 125 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. The other alert is triggered when the temperature exceeds a predefined threshold. This second alert needs to be routed to SMI or SCI based on SW programming.

The PCH in general intends to incorporate multiple thermal sensors for various reasons. The PCH provides a direct reading of the temperature to the host software, to Intel CSME, external EC through the eSPI and any other IP on the PCH which subscribes to receive periodical temperature reporting.

The thermal sensor is operational in S0, Sx and S0ix states.

The Thermal functions working during Sx would be DTS sending periodic temperature to PCH.

The DTS will continue to read temperatures in S0ix modes. However, the frequency of the reads will be less (i.e. once every second). In S0ix mode the DTS has capability to compare the temperature with a temperature threshold (5 º C) programmed value. If the temperature is below the threshold then the DTS will wake up the system and send the temperature to the PCH. Else it will silently discard the temperature and will not send them to the PCH.

Typically the programmed value will be set in the range of 110 ºC to 125 ºC, depending on particular OEM system parameters.

36.2 Temperature Trip Point

The internal thermal sensors 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.

36.3 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.

36.4 Thermal Reporting to 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, Power Management 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.

36.5 Thermal Trip Signal (PCHHOT#)

The PCH provides PCHHOT# signal to indicate that it has exceeded some temperature limit. The BIOS can configure the threshold over which PCHHOT# will assert and sets the PCHHOT# enable to begin reporting. The temperature limit is compared to the present temperature. If the present temperature is greater than the programmed 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 sensors against the programmed temperature limit every 1 second.

NOTE

The "PCHHOT#" on the SOC accounts for both the SOC South and the IOE tiles. In above summary, it mentions PCHHOT# pin only exists in PCH. The PCHHOT# temperature is computed only as a function of the PCH temperature. The processor does not have a PCHHOT# pin.

37.0 Electrical Characteristics

For more information on electrical characteristics, download the pdf, click on the navigation pane and refer the spreadsheet, **833778-001_Electrical_Specification.xlsx**.