



# **Intel<sup>®</sup> 400 Series Chipset Family Platform Controller Hub**

**Datasheet, Volume 1 of 2**

---

***Rev. 003***

***December 2021***



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.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software or service activation.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visit <http://www.intel.com/design/literature.htm>.

Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

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

Copyright © –2021, Intel Corporation. All rights reserved.

## Contents

---

<b>Revision History PCH H.....</b>	<b>14</b>
<b>1.0 Introduction.....</b>	<b>15</b>
1.1 Overview.....	15
1.2 PCH SKUs.....	16
<b>2.0 PCH Controller Device IDs.....</b>	<b>23</b>
2.1 Device and Revision ID Table.....	23
<b>3.0 Flexible I/O.....</b>	<b>26</b>
3.1 Flexible I/O Implementation.....	26
3.1.1 PCH-H.....	26
3.1.2 Flexible I/O Lane Selection.....	27
<b>4.0 Memory Mapping.....</b>	<b>28</b>
4.1 Functional Description.....	28
4.1.1 PCI Devices and Functions.....	28
4.1.2 Fixed I/O Address Ranges.....	30
4.1.3 Variable I/O Decode Ranges.....	33
4.2 Memory Map.....	34
4.2.1 Boot Block Update Scheme.....	36
<b>5.0 System Management.....</b>	<b>38</b>
5.1 Features.....	38
5.1.1 Theory of Operation.....	38
5.1.2 TCO Modes.....	39
<b>6.0 High Precision Event Timer (HPET).....</b>	<b>42</b>
6.1 Timer Accuracy.....	42
6.2 Timer Off-load.....	43
6.3 Interrupt Mapping.....	44
6.4 Periodic Versus Non-Periodic Modes.....	45
6.5 Enabling the Timers.....	47
6.6 Interrupt Levels.....	47
<b>7.0 Thermal Management.....</b>	<b>48</b>
7.1 PCH Thermal Sensor.....	48
7.1.1 Modes of Operation.....	48
7.1.2 Temperature Trip Point.....	48
7.1.3 Thermal Sensor Accuracy $T_{\text{accuracy}}$ .....	48
7.1.4 Thermal Reporting to an EC.....	49
7.1.5 Thermal Trip Signal (PCHHOT#).....	49
<b>8.0 Power and Ground Signals.....</b>	<b>50</b>
<b>9.0 Electrical and Thermal Characteristics.....</b>	<b>52</b>
9.1 Power Rail Absolute Maximum Ratings.....	52
9.2 Thermal Specification.....	52
9.3 General DC Characteristics.....	53
9.4 AC Characteristics.....	68
9.4.1 Panel Power Sequencing and Backlight Control.....	71

9.5 Overshoot/Undershoot Guidelines.....	94
<b>10.0 Pin Straps.....</b>	<b>96</b>
<b>11.0 8254 Timers.....</b>	<b>100</b>
11.1 Timer Programming.....	100
11.2 Reading From Interval Timer.....	101
<b>12.0 Audio Video and Speech.....</b>	<b>103</b>
12.1 Signal Description.....	104
12.2 Integrated Pull-Ups and Pull-Downs.....	106
12.3 I/O Signal Planes and States.....	106
12.4 AVS Feature Summary.....	107
12.4.1 Intel® High Definition Audio Controller Capabilities.....	108
12.4.2 Audio DSP Capabilities.....	108
12.4.3 Intel® High Definition Audio Link Capabilities.....	109
12.4.4 Intel® Display Audio Link Capabilities.....	109
12.4.5 DMIC Interface.....	109
12.4.6 I <sup>2</sup> S/PCM Interface.....	109
12.4.7 SoundWire* Interface.....	109
<b>13.0 Controller Link.....</b>	<b>110</b>
13.1 Signal Description.....	110
13.2 Integrated Pull-Ups and Pull-Downs.....	110
13.3 I/O Signal Planes and States.....	110
13.4 External CL_RST# Pin Driven/Open-drain Mode Support.....	110
<b>14.0 Processor Sideband Signals.....</b>	<b>112</b>
14.1 Signal Description.....	112
14.2 I/O Signal Planes and States.....	112
14.3 Functional Description.....	113
<b>15.0 Digital Display Signals.....</b>	<b>114</b>
15.1 Signal Description.....	114
15.2 Embedded DisplayPort* (eDP*) Backlight Control Signals.....	115
15.3 Integrated Pull-Ups and Pull-Downs.....	115
15.4 I/O Signal Planes and States.....	115
<b>16.0 enhanced Serial Peripheral Interface (eSPI).....</b>	<b>117</b>
16.1 Signal Description.....	117
16.2 Integrated Pull-Ups and Pull-Downs.....	118
16.3 I/O Signal Planes and States.....	118
16.4 Functional Description.....	118
16.4.1 Features.....	118
16.4.2 Protocols.....	119
16.4.3 WAIT States from eSPI Slave.....	119
16.4.4 In-Band Link Reset.....	120
16.4.5 Slave Discovery.....	120
16.4.6 Flash Sharing Mode.....	120
16.4.7 Channels and Supported Transactions.....	120
<b>17.0 General Purpose Input and Output (GPIO).....</b>	<b>127</b>
17.1 Signal Description.....	128

17.2 Functional Description.....	128
17.2.1 GPIO Voltage.....	128
17.2.2 Programmable Hardware Debouncer.....	128
17.2.3 Integrated Pull-Ups and Pull-Downs.....	129
17.2.4 GPPJ_RCOMP_1P8 Signal.....	129
17.2.5 Interrupt / IRQ via GPIO Requirement.....	129
17.2.6 SMI#/SCI and NMI.....	129
17.2.7 Timed GPIO.....	129
17.2.8 GPIO Blink (BK) and Serial Blink (SBK).....	130
17.2.9 GPIO Ownership.....	130
<b>18.0 Intel® Serial I/O Inter-Integrated Circuit (I<sup>2</sup>C) Controllers.....</b>	<b>131</b>
18.1 Signal Description.....	131
18.2 I/O Signal Planes and States.....	132
18.3 Functional Description.....	132
18.3.1 Features.....	132
18.3.2 Protocols Overview.....	133
18.3.3 DMA Controller.....	134
18.3.4 Reset.....	134
18.3.5 Power Management.....	135
18.3.6 Interrupts.....	135
18.3.7 Error Handling.....	136
18.3.8 Programmable SDA Hold Time.....	136
<b>19.0 Gigabit Ethernet Controller.....</b>	<b>137</b>
19.1 Signal Description.....	137
19.2 Integrated Pull-Ups and Pull-Downs.....	138
19.3 I/O Signal Planes and States.....	138
19.4 Functional Description.....	138
19.4.1 GbE PCI Express* Bus Interface.....	140
19.4.2 Error Events and Error Reporting.....	141
19.4.3 Ethernet Interface.....	141
19.4.4 PCI Power Management.....	142
<b>20.0 Interrupt Interface.....</b>	<b>143</b>
20.1 Signal Description.....	143
20.2 Integrated Pull-Ups and Pull-Downs.....	143
20.3 I/O Signal Planes and States.....	143
20.4 Functional Description.....	143
20.4.1 8259 Interrupt Controllers (PIC).....	146
20.4.2 Interrupt Handling.....	147
20.4.3 Initialization Command Words (ICWx).....	149
20.4.4 Operation Command Words (OCW).....	150
20.4.5 Modes of Operation.....	150
20.4.6 Masking Interrupts.....	152
20.4.7 Steering PCI Interrupts.....	152
20.5 Advanced Programmable Interrupt Controller (APIC) (D31:F0).....	153
20.5.1 Interrupt Handling.....	153
20.5.2 Interrupt Mapping.....	153
20.5.3 PCI/PCI Express* Message-Based Interrupts.....	153
20.5.4 IOxAPIC Address Remapping.....	153
20.5.5 External Interrupt Controller Support.....	154

20.6 Serial Interrupt .....	154
20.6.1 Start Frame.....	154
20.6.2 Stop Frame.....	155
20.6.3 Specific Interrupts Not Supported Using SERIRQ.....	155
<b>21.0 Integrated Sensor Hub (ISH).....</b>	<b>157</b>
21.1 Signal Description.....	158
21.2 Integrated Pull-Ups and Pull-Downs.....	159
21.3 I/O Signal Planes and States.....	159
21.4 Functional Description.....	159
21.4.1 ISH Micro-Controller.....	160
21.4.2 SRAM.....	160
21.4.3 PCI Host Interface.....	160
21.4.4 Power Domains and Management .....	161
21.4.5 ISH IPC.....	161
21.4.6 ISH Interrupt Handling via IOAPIC (Interrupt Controller).....	161
21.4.7 ISH I <sup>2</sup> C Controllers.....	162
21.4.8 ISH UART Controller.....	162
21.4.9 ISH GSPI Controller.....	162
21.4.10 ISH GPIOs.....	162
<b>22.0 Low Pin Count (LPC).....</b>	<b>163</b>
22.1 Signal Description.....	164
22.2 Integrated Pull-Ups and Pull-Downs.....	164
22.3 I/O Signal Planes and States.....	164
22.4 Functional Description.....	165
22.4.1 LPC Cycle Types.....	165
22.4.2 Start Field Definition.....	165
22.4.3 Cycle Type/Direction (CYCTYPE + DIR).....	165
22.4.4 Size.....	166
22.4.5 SYNC Timeout.....	166
22.4.6 SYNC Error Indication.....	167
22.4.7 LFRAME# Usage.....	167
22.4.8 I/O Cycles.....	167
22.4.9 LPC Power Management.....	167
22.4.10 Configuration and PCH Implications.....	168
<b>23.0 PCH and System Clocks.....</b>	<b>169</b>
23.1 PCH ICC Clocking Profiles.....	169
23.2 PCH ICC XTAL Input Configurations.....	170
23.3 Signal Descriptions.....	170
23.4 I/O Signal Pin States.....	171
23.5 General Features.....	172
<b>24.0 PCI Express* (PCIe*).....</b>	<b>173</b>
24.1 Signal Description.....	173
24.2 I/O Signal Planes and States.....	174
24.3 PCI Express* Port Support Feature Details.....	174
24.3.1 Intel® Rapid Storage Technology (Intel® RST) for PCIe Storage.....	175
24.3.2 Interrupt Generation.....	176
24.3.3 PCI Express* Power Management.....	176
24.3.4 Dynamic Link Throttling.....	178

24.3.5 Port 8xh Decode.....	178
24.3.6 Separate Reference Clock with Independent SSC (SRIS) .....	179
24.3.7 Advanced Error Reporting.....	179
24.3.8 Single- Root I/O Virtualization (SR- IOV).....	179
24.3.9 SERR# Generation.....	179
24.3.10 Hot-Plug.....	180
24.3.11 PCI Express* Lane Polarity Inversion.....	180
24.3.12 PCI Express* Controller Lane Reversal.....	181
24.3.13 Precision Time Measurement (PTM).....	181
<b>25.0 Power Management.....</b>	<b>182</b>
25.1 Signal Description.....	182
25.2 Integrated Pull-Ups and Pull-Downs.....	185
25.3 I/O Signal Planes and States.....	185
25.4 Functional Description.....	187
25.4.1 Features.....	187
25.4.2 PCH and System Power States.....	188
25.4.3 System Power Planes.....	190
25.4.4 SMI#/SCI Generation.....	191
25.4.5 C-States.....	193
25.4.6 Dynamic 24 MHz Clock Control.....	193
25.4.7 Sleep States.....	194
25.4.8 Event Input Signals and Their Usage.....	199
25.4.9 ALT Access Mode.....	203
25.4.10 System Power Supplies, Planes, and Signals.....	205
25.4.11 Legacy Power Management Theory of Operation.....	209
25.4.12 Reset Behavior.....	209
<b>26.0 Real Time Clock (RTC).....</b>	<b>212</b>
26.1 Signal Description.....	212
26.2 I/O Signal Planes and States.....	213
26.3 Functional Description.....	213
26.3.1 Update Cycles.....	214
26.3.2 Interrupts.....	214
26.3.3 Lockable RAM Ranges.....	214
26.3.4 Century Rollover.....	214
26.3.5 Clearing Battery-Backed RTC RAM.....	215
26.3.6 External RTC Circuitry.....	215
<b>27.0 Serial ATA (SATA).....</b>	<b>216</b>
27.1 Signals Description.....	216
27.2 Integrated Pull-Ups and Pull-Downs.....	219
27.3 I/O Signal Planes and States.....	220
27.4 Functional Description.....	220
27.4.1 SATA 6 Gb/s Support.....	221
27.4.2 SATA Feature Support.....	221
27.4.3 Hot-Plug Operation.....	221
27.4.4 Intel® Rapid Storage Technology (Intel® RST).....	221
27.4.5 Power Management Operation.....	222
27.4.6 SATA Device Presence.....	225
27.4.7 SATA LED.....	225
27.4.8 Advanced Host Controller Interface (AHCI) Operation.....	225

27.4.9 Enclosure Management (SGPIO Signals).....	226
<b>28.0 System Management Interface and SMLink.....</b>	<b>229</b>
28.1 Signal Description.....	229
28.2 Integrated Pull-Ups and Pull-Downs.....	229
28.3 I/O Signal Planes and States.....	230
28.4 Functional Description.....	230
<b>29.0 Host System Management Bus (SMBus) Controller.....</b>	<b>231</b>
29.1 Signal Description.....	231
29.2 Integrated Pull-Ups and Pull-Downs.....	231
29.3 I/O Signal Planes and States.....	232
29.4 Functional Description.....	232
29.4.1 Host Controller.....	232
29.4.2 SMBus Slave Interface.....	239
29.5 SMBus Power Gating.....	246
<b>30.0 Serial Peripheral Interface (SPI).....</b>	<b>247</b>
30.1 Signal Description.....	247
30.2 Integrated Pull-Ups and Pull-Downs.....	248
30.3 I/O Signal Planes and States.....	249
30.4 Functional Description.....	249
30.4.1 SPI0 for Flash.....	249
30.4.2 SPI0 Support for TPM.....	254
30.4.3 SPI1 Support for Touch Device.....	254
<b>31.0 Intel® Serial I/O Generic SPI (GSPI) Controllers.....</b>	<b>255</b>
31.1 Signal Description.....	255
31.2 Integrated Pull-Ups and Pull-Downs.....	256
31.3 I/O Signal Planes and States.....	256
31.4 Functional Description.....	256
31.4.1 Features.....	257
31.4.2 Controller Overview.....	257
31.4.3 DMA Controller.....	257
31.4.4 Reset.....	258
31.4.5 Power Management.....	259
31.4.6 Interrupts.....	259
31.4.7 Error Handling.....	259
<b>32.0 Testability.....</b>	<b>260</b>
32.1 JTAG.....	260
32.1.1 Signal Description.....	260
32.1.2 I/O Signal Planes and States.....	261
32.2 Intel® Trace Hub (Intel® TH).....	261
32.2.1 Platform Setup.....	262
32.3 Direct Connect Interface (DCI).....	262
32.3.1 Out Of Band (OOB) Hosting DCI.....	263
32.3.2 USB 3.2 Gen 1x1 (5 Gb/s) and USB 2.0 Hosting DCI.DBC.....	263
32.3.3 Platform Setup.....	264
<b>33.0 Intel® Serial I/O Universal Asynchronous Receiver/Transmitter (UART) Controllers.....</b>	<b>265</b>
33.1 Signal Description.....	265

33.2 I/O Signal Planes and States.....	266
33.3 Functional Description.....	266
33.3.1 Features.....	266
33.3.2 UART Serial (RS-232) Protocols Overview.....	267
33.3.3 16550 8-bit Addressing - Debug Driver Compatibility.....	268
33.3.4 DMA Controller.....	268
33.3.5 Reset.....	269
33.3.6 Power Management.....	269
33.3.7 Interrupts.....	270
33.3.8 Error Handling.....	270
<b>34.0 Universal Serial Bus (USB).....</b>	<b>271</b>
34.1 Signal Description.....	271
34.2 Integrated Pull-Ups and Pull-Downs.....	275
34.3 I/O Signal Planes and States.....	275
34.4 Functional Description.....	276
34.4.1 eXtensible Host Controller Interface (xHCI) Controller (D20:F0).....	276
34.4.2 USB Dual Role Support - eXtensible Device Controller Interface (xHCI) Controller (D20:F1).....	277
34.4.3 Supported USB 2.0 Ports.....	277
<b>35.0 Connectivity Integrated (CNVi).....</b>	<b>278</b>
35.1 Signal Description.....	278
35.2 Platform PU/PD requirements.....	281
35.3 I/O Signal Planes and States.....	281
35.4 Functional Description.....	282
<b>36.0 Secure Digital eXtended Capacity (SDXC).....</b>	<b>284</b>
36.1 Signal Description.....	284
36.2 I/O Signal Planes and States.....	285
36.3 Functional Description.....	285
<b>37.0 Direct Media Interface (DMI).....</b>	<b>286</b>
37.1 Signal Description.....	286
37.2 I/O Signal Planes and States.....	286
37.3 Functional Description.....	286
37.3.1 Lane Reversal.....	286
37.3.2 Polarity Inversion.....	286
<b>38.0 Private Configuration Space Target Port ID.....</b>	<b>287</b>

## Figures

1	High Speed I/O (HSIO) Lane Multiplexing in PCH-H.....	26
2	TCO Compatible Mode SMBus Configuration.....	39
3	Advanced TCO Mode.....	41
4	PCI Express* Transmitter Eye.....	70
5	PCI Express* Receiver Eye.....	71
6	Panel Power Sequencing.....	72
7	Clock Timing.....	76
8	Measurement Points for Differential Waveforms.....	77
9	SMBus/SMLink Transaction.....	78
10	USB Rise and Fall Times.....	79
11	USB Jitter.....	80
12	USB EOP Width.....	80
13	SMBus/SMLink Timeout.....	82
14	Intel® High Definition Audio (Intel® HD Audio) Input and Output Timings.....	83
15	Valid Delay from Rising Clock Edge.....	83
16	Setup and Hold Times.....	84
17	Float Delay.....	84
18	Output Enable Delay.....	84
19	Pulse Width.....	85
20	PCH Test Load.....	86
21	SPI/GSPI/eSPI Timings.....	91
22	Controller Link Receive Timings.....	92
23	Controller Link Receive Slew Rate.....	93
24	Maximum Acceptable Overshoot/Undershoot Waveform.....	95
25	Basic eSPI Protocol.....	119
26	eSPI Slave Request to PCH for PCH Temperature.....	123
27	PCH Response to eSPI Slave with PCH Temperature.....	123
28	eSPI Slave Request to PCH for PCH RTC Time .....	124
29	PCH Response to eSPI Slave with RTC Time.....	125
30	Data Transfer on the I2C Bus.....	133
31	LPC Interface Diagram.....	163
32	Integrated Clock Controller (ICC) Diagram.....	169
33	Supported PCI Express* Link Configurations.....	174
34	Generation of SERR# to Platform.....	180
35	Conceptual Diagram of SLP_LAN#.....	207
36	Flow for Port Enable/Device Present Bits.....	225
37	Serial Data transmitted over the SGPIO Interface.....	228
38	Flash Descriptor Regions.....	252
39	Platform Setup with Intel® Trace Hub.....	262
40	Platform Setup with DCI Connection.....	264
41	UART Serial Protocol.....	267
42	UART Receiver Serial Data Sample Points.....	268
43	Supported USB 2.0 Ports.....	277

## Tables

1	Desktop PCH SKUs.....	16
2	Workstation PCH SKUs.....	17
3	Desktop PCH HSIO Details.....	18
4	Workstation PCH HSIO Details.....	19
5	Mobile PCH SKUs.....	20
6	Mobile PCH HSIO Details.....	21
7	PCH-H Device and Revision ID Table.....	23
8	PCI Devices and Functions.....	28
9	Fixed I/O Ranges Decoded by PCH.....	30
10	Variable I/O Decode Ranges.....	33
11	PCH Memory Decode Ranges (Processor Perspective).....	34
12	Boot Block Update Scheme.....	36
13	Event Transitions that Cause Messages.....	40
14	Legacy Replacement Routing.....	44
15	Power Rails Descriptions PCH-H.....	50
16	PCH Absolute Power Rail Maximum and Minimum Ratings.....	52
17	Operating Junction Temperature Range.....	52
18	PCH Icc <sup>3</sup> with Integrated 1.8 V VRM Mode OFF (Desktop SKUs).....	53
19	PCH Icc <sup>3</sup> with Integrated 1.8 V VRM Mode OFF (H Mobile SKUs).....	54
20	PCH Icc <sup>3</sup> with Integrated 1.8 V VRM Mode ON (Desktop SKUs).....	55
21	PCH Icc <sup>3</sup> with Integrated 1.8 V VRM Mode ON (H Mobile SKUs).....	56
22	PCH VCCPRIM_1P05 Adder per HSIO Lane.....	57
23	PCH VCCPRIM_1P8 Adder with and without CNVi .....	57
24	PCH VCCPRIM_3P3 Adder with and without CNVi.....	58
25	Single-Ended Signal DC Characteristics as Inputs or Outputs.....	58
26	Single-Ended Signal DC Characteristics as Inputs or Outputs.....	64
27	Differential Signals Characteristics.....	64
28	Power Rail Operational Voltage Range.....	67
29	PCI Express* Interface Timings.....	68
30	DDC Characteristics.....	71
31	DisplayPort* Hot-Plug Detect Interface.....	72
32	Clock Timings.....	72
33	USB 2.0 Timing.....	78
34	USB 3.2 Interface Transmit and Receiver Timings.....	79
35	SATA Interface Timings.....	80
36	SMBus and SMLink Timing.....	81
37	Intel® High Definition Audio (Intel® HD Audio) Timing.....	82
38	LPC Timing (24 MHz).....	83
39	Miscellaneous Timings.....	85
40	SPI Timings (20 MHz).....	86
41	SPI Timings (33 MHz).....	87
42	SPI Timings (50 MHz).....	87
43	SPI Touch Timings (20 MHz).....	88
44	SPI Touch Timings (33 MHz) 1.8 V/3.3 V.....	89
45	eSPI Timings (33 MHz).....	89
46	eSPI Timings (50 MHz).....	90
47	eSPI Timings (66 MHz).....	90
48	GSPI Timings (25 MHz).....	92
49	Controller Link Receive Timings.....	92
50	UART Timings.....	93
51	I <sup>2</sup> S Timings - Master Mode.....	93
52	I <sup>2</sup> S Timing - Slave Mode (non S0ix).....	93
53	I <sup>2</sup> S Timing - Slave Mode (S0ix).....	94
54	3.3 V Overshoot/Undershoot Specifications.....	94

55	Pin Straps.....	96
56	Counter Operating Modes.....	101
57	Digital Display Signals.....	114
58	eSPI Channels and Supported Transactions.....	120
59	eSPI Virtual Wires (VW).....	121
60	GPIO Group Summary.....	127
61	GbE LAN Signals.....	137
62	Power Plane and States for Output Signals .....	138
63	Power Plane and States for Input Signals .....	138
64	LAN Mode Support .....	142
65	Interrupt Options - 8259 Mode.....	144
66	Interrupt Options - APIC Mode .....	145
67	Interrupt Logic Signals.....	146
68	Interrupt Controllers PIC .....	146
69	Interrupt Status Registers.....	147
70	Content of Interrupt Vector Byte .....	148
71	Stop Frame Explanation.....	155
72	Data Frame Format .....	156
73	IPC Initiator -> Target Flows.....	161
74	LPC Cycle Types Supported .....	165
75	Cycle Type Bit Definitions.....	166
76	Transfer Size Bit Definition.....	166
77	SYNC Bit Definition .....	166
78	PCI Express* Port Feature Details.....	174
79	MSI Versus PCI IRQ Actions .....	176
80	General Power States for Systems Using the PCH.....	188
81	State Transition Rules for the PCH.....	189
82	System Power Plane.....	190
83	Causes of SMI and SCI.....	191
84	Sleep Types.....	195
85	Causes of Wake Events.....	195
86	Transitions Due to Power Failure.....	197
87	Supported Deep Sx Policy Configurations.....	198
88	Deep Sx Wake Events.....	199
89	Transitions Due to Power Button.....	200
90	Write Only Registers with Read Paths in ALT Access Mode.....	204
91	PIC Reserved Bits Return Values.....	204
92	SUSPWRDNACK/SUSWARN#/GPP_A13 Pin Behavior.....	208
93	SUSPWRDNACK During Reset.....	208
94	Causes of Host and Global Resets.....	210
95	RTC Crystal Requirements .....	215
96	External Crystal Oscillator Requirements .....	215
97	I <sup>2</sup> C Block Read.....	235
98	Enable for SMBALERT#.....	238
99	Enables for SMBus Slave Write and SMBus Host Events.....	238
100	Enables for the Host Notify Command.....	238
101	Slave Write Registers.....	239
102	Command Types.....	240
103	Slave Read Cycle Format.....	240
104	Data Values for Slave Read Registers.....	241
105	Host Notify Format.....	243
106	Slave Read Cycle Format.....	244
107	Data Values for Slave Read Registers.....	244
108	Enables for SMBus Slave Write and SMBus Host Events.....	246
109	SPI0 Flash Regions.....	250

110	Region Size Versus Erase Granularity of Flash Components.....	251
111	Region Access Control Table.....	253
112	Testability Signals.....	260
113	Power Planes and States for Testability Signals.....	261
114	SD Working Modes.....	285
115	Private Configuration Space Register Target Port IDs.....	287

## Revision History PCH H

Document Number	Revision Number	Revision Description	Date
620854	001	Initial Release	April 2020
620854`	002	<p><a href="#">Introduction</a> on page 15</p> <ul style="list-style-type: none"> <li>• <b>Added</b> Z490, H470, and Q470 desktop SKUs and W480 Workstation SKUs</li> </ul> <p><a href="#">Flexible I/O</a> on page 26</p> <ul style="list-style-type: none"> <li>• <b>Updated</b> <a href="#">Figure 1</a> on page 26</li> </ul> <p><a href="#">Electrical and Thermal Characteristics</a> on page 52</p> <ul style="list-style-type: none"> <li>• <b>Updated</b> <a href="#">Table 38</a> on page 83</li> </ul>	May 2020
620854	003	<p><a href="#">Testability</a> on page 260</p> <ul style="list-style-type: none"> <li>• Updated Content in <a href="#">Table 112</a></li> </ul> <p><a href="#">Private Configuration Space Target Port ID</a> on page 287</p> <ul style="list-style-type: none"> <li>• Updated DCI port ID in <a href="#">Table 115</a></li> </ul>	December 2021

## 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® 400 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 SKUs, unless specifically noted otherwise.
- Throughout this document, PCH refers to desktop, server/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 “Server/Workstation” and “Server/Workstation Only” refers to information that is applicable only to Server/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 manual assumes a working knowledge of the vocabulary and principles of interfaces and architectures such as PCI Express\* (PCIe\*), Universal Serial Bus (USB), Advance Host Controller Interface (AHCI), eXtensible Host Controller Interface (xHCI), and so on. This manual abbreviates buses as *B<sub>n</sub>*, devices as *D<sub>n</sub>* and functions as *F<sub>n</sub>*. For example Device 31 Function 0 is abbreviated as D31:F0, Bus 1 Device 8 Function 0 is abbreviated as B1:D8:F0. Generally, the bus number will not be used, and can be considered to be Bus 0.

### References

Specification	Document Number
Intel® 400 Series Chipset Family Platform Controller Hub Datasheet, Volume 2 of 2	620855

## 1.1 Overview

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

- ACPI Power Management Logic Support, Revision 4.0a
- PCI Express\* Base Specification Revision 3.0
- Integrated Serial ATA Host Controller 3.2
- USB 3.2 Gen 2x1 (10 Gb/s) eXtensible Host Controller (xHCI)
- USB 3.2 Gen 1x1 (5 Gb/s) Dual Role (eXtensible Device Controller - xDCI) Capability
- Direct Media Interface (DMI)
- Serial Peripheral Interface (SPI)

- Enhanced Serial Peripheral Interface (eSPI)
- Flexible I/O—Allows some high speed I/O signals to be configured as PCIe\*, SATA 6 Gb/s or USB 3.2
- General Purpose Input Output (GPIO)
- Low Pin Count (LPC) interface
- Interrupt Controller
- Timer functions
- System Management Bus (SMBus) Specification, Version 2.0
- Integrated Clock Controller (ICC) / Real Time Clock (RTC)
- Intel® High Definition Audio and Intel® Smart Sound Technology (Intel® SST) supporting Intel® HD Audio, I<sup>2</sup>S, MIPI SoundWire\*, and DMIC interfaces
- Intel® Serial I/O UART Host Controllers
- Intel® Serial I/O I<sup>2</sup>C host Controllers
- Integrated 10/100/1000 Gigabit Ethernet MAC
- Integrated Sensor Hub (ISH)
- Supports Intel® Rapid Storage Technology (Intel® RST)
- Supports Intel® Active Management Technology (Intel® AMT)
- Supports Intel® Virtualization Technology for Directed I/O (Intel® VT-d)
- Supports Intel® Trusted Execution Technology (Intel® TXT)
- JTAG Boundary Scan support
- Intel® Trace Hub (Intel® TH) and Direct Connect Interface (DCI) for debug
- Supports Intel® Converged Security and Management Engine (Intel® CSME) with firmware version 14
- Integrated Intel® Wireless-AX Support

**NOTE**

Not all functions and capabilities may be available on all SKUs.

## 1.2 PCH SKUs

The following table provides an overview of the PCH I/O capabilities.

**Table 1. Desktop PCH SKUs**

Features	SKU		
	Z490	H470	Q470
DMI	DMI 3.0 x4	DMI 3.0 x4	DMI 3.0 x4
SATA 6 Gb/s Ports	Up to 6	Up to 6	Up to 6
PCIe*	Up to 24 Gen 3 lanes	Up to 20 Gen 3 lanes	Up to 24 Gen 3 lanes
<i>continued...</i>			

Features	SKU		
	Z490	H470	Q470
Total USB Ports (Maximum USB 3.2 Gen 1x1 (5 Gb/s))	14 (10)	14 (8)	14 (10)
Maximum USB 3.2 Ports: USB 3.2 Gen 2x1 (10 Gb/s) / USB 3.2 Gen 1x1 (5 Gb/s)	6 / 10	4 / 8	6 / 10
Total USB 2.0 Ports	14	14	14
Intel® Smart Sound Technology	YES	YES	YES
Intel® CSME 14 Firmware	Consumer	Consumer	Corporate
Intel® AMT	NO	NO	YES
Intel® Optane™ Memory Support	YES	YES	YES
Intel® Rapid Storage Technology	YES	YES	YES
Intel® RST for PCIe* Storage Ports <sup>1</sup>	3	2	3
Intel® RST RAID Support	YES	YES	YES
eSPI Chip Select	2	2	2
Intel® Trusted Execution Technology	NO	NO	YES
Simultaneous Independent Display Support <sup>2</sup>	3	3	3
Notes: 1. Refer to <a href="#">Intel® Rapid Storage Technology (Intel® RST) for PCIe Storage</a> on page 175 for more information on the Intel® RST for PCIe* Storage port locations. 2. This is applicable to integrated graphics only.			

**Table 2. Workstation PCH SKUs**

Features	SKU
	W480
DMI	DMI 3.0 x4
SATA 6 Gb/s Ports	Up to 8
PCIe*	Up to 24 Gen 3 lanes
Total USB Ports (Maximum USB 3.2 Gen 1x1 (5 Gb/s))	14 (10)
Maximum USB 3.2 Ports: USB 3.2 Gen 2x1 (10 Gb/s) / USB 3.2 Gen 1x1 (5 Gb/s)	8 / 10
Total USB 2.0 Ports	14
Intel® Smart Sound Technology	YES
Intel® CSME 14 Firmware	Corporate
Intel® AMT	YES
Intel® Optane™ Memory Support	YES
Intel® Rapid Storage Technology	YES
Intel® RST for PCIe* Storage Ports <sup>1</sup>	3
Intel® RST RAID Support	YES
eSPI Chip Select	2
<i>continued...</i>	

Features	SKU
	W480
Intel® Trusted Execution Technology	YES
Simultaneous Independent Display Support <sup>2</sup>	3
Notes: 1. Refer to <a href="#">Intel® Rapid Storage Technology (Intel® RST) for PCIe Storage</a> on page 175 for more information on the Intel® RST for PCIe* Storage port locations. 2. This is applicable to integrated graphics only	

**Table 3. Desktop PCH HSIO Details**

Flex I/O Lane	SKU		
	Z490	H470	Q470
0	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)
1	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)
2	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)
3	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)
4	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 1x1 (5 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)
5	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 1x1 (5 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)
6	PCIe;USB 3.2 Gen 1x1 (5 Gb/s)	USB 3.2 Gen 1x1 (5 Gb/s)	PCIe;USB 3.2 Gen 1x1 (5 Gb/s)
7	PCIe;USB 3.2 Gen 1x1 (5 Gb/s)	USB 3.2 Gen 1x1 (5 Gb/s)	PCIe;USB 3.2 Gen 1x1 (5 Gb/s)
8	PCIe;USB 3.2 Gen 1x1 (5 Gb/s)	PCIe	PCIe;USB 3.2 Gen 1x1 (5 Gb/s)
9	PCIe;USB 3.2 Gen 1x1 (5 Gb/s)	PCIe	PCIe;USB 3.2 Gen 1x1 (5 Gb/s)
10	PCIe;GbE	PCIe;GbE	PCIe;GbE
11	PCIe	PCIe	PCIe
12	PCIe	PCIe	PCIe
13	PCIe	PCIe	PCIe
14	PCIe;GbE	PCIe;GbE	PCIe;GbE
15	PCIe	PCIe	PCIe
16	PCIe;SATA 0A	PCIe;SATA 0A	PCIe;SATA 0A
17	SATA 1A;PCIe;GbE	SATA 1A;PCIe;GbE	SATA 1A;PCIe;GbE
18	SATA 0B;PCIe;GbE	SATA 0B;PCIe;GbE	SATA 0B;PCIe;GbE
19	SATA 1B;PCIe	SATA 1B;PCIe	SATA 1B;PCIe
20	SATA 2;PCIe	SATA 2;PCIe	SATA 2;PCIe
21	SATA 3;PCIe	SATA 3;PCIe	SATA 3;PCIe
22	SATA 4;PCIe	SATA 4	SATA 4;PCIe
23	SATA 5;PCIe	SATA 5	SATA 5;PCIe
24	PCIe	PCIe	PCIe
25	PCIe	PCIe	PCIe
26	PCIe	PCIe	PCIe
<b>continued...</b>			

Flex I/O Lane	SKU		
	Z490	H470	Q470
27	PCIe	PCIe	PCIe
28	PCIe	PCIe	PCIe
29	PCIe	PCIe	PCIe

**Table 4. Workstation PCH HSIO Details**

Flex I/O Lane	SKU
	W480
0	USB 3.2 Gen 2x1 (10 Gb/s)
1	USB 3.2 Gen 2x1 (10 Gb/s)
2	USB 3.2 Gen 2x1 (10 Gb/s)
3	USB 3.2 Gen 2x1 (10 Gb/s)
4	USB 3.2 Gen 2x1 (10 Gb/s)
5	USB 3.2 Gen 2x1 (10 Gb/s)
6	PCIe*;USB 3.2 Gen 2x1 (10 Gb/s)
7	PCIe;USB 3.2 Gen 2x1 (10 Gb/s)
8	PCIe;USB 3.2 Gen 1x1 (5 Gb/s)
9	PCIe;USB 3.2 Gen 1x1 (5 Gb/s)
10	PCIe;GbE
11	PCIe
12	PCIe
13	PCIe
14	GbE ; PCIe / Intel® RST for PCIe Storage Port
15	PCIe / Intel® RST for PCIe Storage Port
16	SATA 0A / PCIe / Intel® RST for PCIe Storage Port
17	SATA 1A;GbE / PCIe / Intel® RST for PCIe Storage Port
18	SATA 0B;PCIe;GbE
19	SATA 1B;PCIe
20	SATA 2;PCIe
21	SATA 3;PCIe
22	SATA 4;PCIe / Intel® RST for PCIe Storage Port
23	SATA 5;PCIe / Intel® RST for PCIe Storage Port
24	SATA 6;PCIe / Intel® RST for PCIe Storage Port
25	SATA 7;PCIe / Intel® RST for PCIe Storage Port
26	PCIe / Intel® RST for PCIe Storage Port
continued...	

Flex I/O Lane	SKU
	W480
27	PCIe / Intel® RST for PCIe Storage Port
28	PCIe / Intel® RST for PCIe Storage Port
29	PCIe / Intel® RST for PCIe Storage Port

**Table 5. Mobile PCH SKUs**

Features	SKU		
	HM470	QM480	WM490
DMI	DMI 3.0 x4	DMI 3.0 x4	DMI 3.0 x4
SATA 6 Gb/s Ports	Up to 4	Up to 4	Up to 8
PCIe*	Up to 16 Gen 3 lanes	Up to 20 Gen 3 lanes	Up to 24 Gen 3 lanes
Total USB Ports (Maximum USB 3.2 Gen 1x1 (5 Gb/s)	14 (8)	14 (10)	14 (10)
Maximum USB 3.2 Ports: USB 3.2 Gen 2x1 (10 Gb/s) / USB 3.2 Gen 1x1 (5 Gb/s)	4 / 8	6 / 10	6 / 10
Total USB 2.0 Ports	14	14	14
Intel® Smart Sound Technology	YES	YES	YES
Intel® CSME 14 Firmware	Consumer	Consumer / Corporate	Consumer / Corporate
Intel® AMT	NO	YES	YES
Intel® Optane™ Memory Support	YES	YES	YES
Intel® Rapid Storage Technology	YES	YES	YES
Intel® RST for PCIe* Storage Ports <sup>1</sup>	2	2	3
Intel® RST RAID Support	YES	YES	YES
eSPI Chip Select	1	1	1
Intel® Trusted Execution Technology	NO	YES	YES
Simultaneous Independent Display Support <sup>2</sup>	3	3	3
<p><i>Notes:</i> 1. Refer to <a href="#">Intel® Rapid Storage Technology (Intel® RST) for PCIe Storage</a> on page 175 for more information on the Intel® RST for PCIe* Storage port locations.</p> <p>2. This is applicable to integrated graphics only.</p>			

**Table 6. Mobile PCH HSIO Details**

Flex I/O Lane	SKU		
	HM470	QM480	WM490
0	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)
1	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)
2	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)
3	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)
4	USB 3.2 Gen 1x1 (5 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)
5	USB 3.2 Gen 1x1 (5 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)	USB 3.2 Gen 2x1 (10 Gb/s)
6	USB 3.2 Gen 1x1 (5 Gb/s)	USB 3.2 Gen 1x1 (5 Gb/s)	PCIe*;USB 3.2 Gen 1x1 (5 Gb/s)
7	USB 3.2 Gen 1x1 (5 Gb/s)	USB 3.2 Gen 1x1 (5 Gb/s)	PCIe;USB 3.2 Gen 1x1 (5 Gb/s)
8	N/A	USB 3.2 Gen 1x1 (5 Gb/s)	PCIe;USB 3.2 Gen 1x1 (5 Gb/s)
9	N/A	USB 3.2 Gen 1x1 (5 Gb/s)	PCIe;USB 3.2 Gen 1x1 (5 Gb/s)
10	GbE	PCIe;GbE	PCIe;GbE
11	N/A	PCIe	PCIe
12	PCIe	PCIe	PCIe
13	N/A	PCIe	PCIe
14	GbE ; PCIe / Intel® RST for PCIe Storage Port	GbE ; PCIe / Intel® RST for PCIe Storage Port	GbE ; PCIe / Intel® RST for PCIe Storage Port
15	PCIe / Intel® RST for PCIe Storage Port	PCIe / Intel® RST for PCIe Storage Port	PCIe / Intel® RST for PCIe Storage Port
16	SATA 0A ;PCIe / Intel® RST for PCIe Storage Port	SATA 0A ;PCIe / Intel® RST for PCIe Storage Port	SATA 0A ;PCIe / Intel® RST for PCIe Storage Port
17	SATA 1A ; GbE ; PCIe / Intel® RST for PCIe Storage Port	SATA 1A ; GbE ; PCIe / Intel® RST for PCIe Storage Port	SATA 1A ; GbE ; PCIe / Intel® RST for PCIe Storage Port
18	SATA 0B;PCIe;GbE	SATA 0B;PCIe;GbE	SATA 0B;PCIe;GbE
19	SATA 1B;PCIe	SATA 1B;PCIe	SATA 1B;PCIe
20	PCIe	PCIe	SATA 2;PCIe
21	PCIe	PCIe	SATA 3;PCIe
22	SATA 4;PCIe	SATA 4;PCIe	SATA 4;PCIe / Intel® RST for PCIe Storage Port
23	SATA 5;PCIe	SATA 5;PCIe	SATA 5;PCIe / Intel® RST for PCIe Storage Port
24	PCIe	PCIe	SATA 6;PCIe / Intel® RST for PCIe Storage Port
25	PCIe	PCIe	SATA 7;PCIe / Intel® RST for PCIe Storage Port
26	PCIe / Intel® RST for PCIe Storage Port	PCIe / Intel® RST for PCIe Storage Port	PCIe / Intel® RST for PCIe Storage Port

*continued...*

Flex I/O Lane	SKU		
	HM470	QM480	WM490
27	PCIe / Intel® RST for PCIe Storage Port	PCIe / Intel® RST for PCIe Storage Port	PCIe / Intel® RST for PCIe Storage Port
28	PCIe / Intel® RST for PCIe Storage Port	PCIe / Intel® RST for PCIe Storage Port	PCIe / Intel® RST for PCIe Storage Port
29	PCIe / Intel® RST for PCIe Storage Port	PCIe / Intel® RST for PCIe Storage Port	PCIe / Intel® RST for PCIe Storage Port

## 2.0 PCH Controller Device IDs

This chapter contains information about the Device and Revision ID table.

### 2.1 Device and Revision ID Table

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

**Table 7. PCH-H Device and Revision ID Table**

Device ID	Device Function - Device Description	A0 SRID	Note
0680 - 069F	LPC Controller (eSPI Enable Strap = 0)/ eSPI Controller (eSPI Enable Strap = 1)	00	<b>PCH Device IDs</b> <b>Desktop SKUs</b> H470:0684 Z490:0685 Q470:0687 <b>Mobile SKUs</b> HM470:068D QM480:068C WM490:068E <b>Workstation SKU</b> W480:0697
06A0	D31:F1 - P2SB	00	
06A1	D31:F2 - PMC	00	
06A3	D31:F4 - SMBus	00	
06A4	D31:F5 - SPI (flash) Controller	00	
0D4C	D31:F6 - GbE Controller: Corporate/ vPro (Default)	00	
0D4D	D31:F6 - GbE Controller: Consumer	00	
06A6	D31:F7 - Intel® Trace Hub	00	
06A8	D30:F0 - UART #0	00	
06A9	D30:F1 - UART #1	00	
06AA	D30:F2 - GSPI #0	00	
06AB	D30:F3 - GSPI #1	00	
06AC	D27:F4 - PCI Express Root Port #21	F0	
06AD	D27:F5 - PCI Express Root Port #22	F0	
06AE	D27:F6 - PCI Express Root Port #23	F0	
06AF	D27:F7 - PCI Express Root Port #24	F0	
06B0	D29:F0 - PCI Express Root Port #9	F0	
continued...			

Device ID	Device Function - Device Description	A0 SRID	Note
06B1	D29:F1 - PCI Express Root Port #10	F0	
06B2	D29:F2 - PCI Express Root Port #11	F0	
06B3	D29:F3 - PCI Express Root Port #12	F0	
06B4	D29:F4 - PCI Express Root Port #13	F0	
06B5	D29:F5 - PCI Express Root Port #14	F0	
06B6	D29:F6 - PCI Express Root Port #15	F0	
06B7	D29:F7 - PCI Express Root Port #16	F0	
06B8	D28:F0 - PCI Express Root Port #1	F0	
06B9	D28:F1 - PCI Express Root Port #2	F0	
06BA	D28:F2 - PCI Express Root Port #3	F0	
06BB	D28:F3 - PCI Express Root Port #4	F0	
06BC	D28:F4 - PCI Express Root Port #5	F0	
06BD	D28:F5 - PCI Express Root Port #6	F0	
06BE	D28:F6 - PCI Express Root Port #7	F0	
06BF	D28:F7 - PCI Express Root Port #8	F0	
06C0	D27:F0 - PCI Express Root Port #17	F0	
06C1	D27:F1 - PCI Express Root Port #18	F0	
06C2	D27:F2 - PCI Express Root Port #19	F0	
06C3	D27:F3 - PCI Express Root Port #20	F0	
06C7	D25:F2 - UART #2	00	
06C8	D31:F3 - cAVS 1.8	00	
06D2	D23:F0 - SATA Controller (AHCI) - Desktop / Workstation	00	Desktop / Workstation SKUs: Z490,H470, Q470,W480
06D3	D23:F0 - SATA Controller (AHCI) - Mobile Halo	00	Mobile Halo SKUs: HM470,QM480,WM490
06D5	D23:F0 - SATA Controller (RAID 0/1/5/10) - NOT premium - Mobile Halo	00	
06D7	D23:F0 - SATA Controller (RAID 0/1/5/10) - premium - Mobile Halo	00	Premium - Mobile Halo SKUs: HM470,QM480,WM490
2822	D23:F0 - SATA Controller (RAID 0/1/5/10) - In-box Compatible ID (Intel® RST driver) - Desktop / Workstation	00	Desktop / Workstation SKUs: Z490,H470,Q470,W480
282A	D23:F0 - SATA Controller (RAID 0/1/5/10) - In-box Compatible ID (Intel® RST driver) - Mobile Halo	00	Mobile Halo SKUs: HM470,QM480,WM490
06DE	D23:F0 - SATA Controller (AHCI) Optane Caching - Desktop	00	Desktop SKUs: Z490,H470,Q470
06E0	D22:F0 - HECI #1	00	
06E1	D22:F1 - HECI #2	00	
<b>continued...</b>			

Device ID	Device Function - Device Description	A0 SRID	Note
06E2	D22:F2 - IDE Redirection (IDE-R)	00	
06E3	D22:F3 - Keyboard and Text (KT) Redirection	00	
06E4	D22:F4 - HECI #3	00	
06E5	D22:F5 - HECI #4	00	
06E8	D21:F0 - I <sup>2</sup> C Controller #0	00	
06E9	D21:F1 - I <sup>2</sup> C Controller #1	00	
06EA	D21:F2 - I <sup>2</sup> C Controller #2	00	
06EB	D21:F3 - I <sup>2</sup> C Controller #3	00	
06ED	D20:F0 - USB 3.2 Gen 2x1 (10 Gb/s) xHCI HC	00	
06EF	D20:F2 - Shared SRAM	00	
06F0	D20:F3 - CNVi: WiFi	00	
06F5	D20:F5 - SCS3: SDXC	00	
06F9	D18:F0 - Thermal Subsystem	00	
06FB	D18:F6 - GSPI #2	00	
06FC	D19:F0 - Integrated Sensor Hub	00	

## 3.0 Flexible I/O

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

### NOTE

Some Flexible I/O multiplexing capabilities are not available on all SKUs.

## 3.1 Flexible I/O Implementation

Topics Covered:

- PCH-H
- Flexible I/O Lane Selection

### 3.1.1 PCH-H

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

Flex I/O Lane	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
High Speed I/O (HSIO) Type and Lane	USB 3.2 Gen 1x1/2x1 #1	USB 3.2 Gen 1x1/2x1 #2	USB 3.2 Gen 1x1/2x1 #3	USB 3.2 Gen 1x1/2x1 #4	USB 3.2 Gen 1x1/2x1 #5	USB 3.2 Gen 1x1/2x1 #6	USB 3.2 Gen 1x1/2x1 #7	USB 3.2 Gen 1x1/2x1 #8	USB 3.2 Gen 1x1/2x1 #9	USB 3.2 Gen 1x1/2x1 #10	PCIe* #5	PCIe* #6	PCIe* #7	PCIe* #8	PCIe* #9	PCIe* #10	PCIe* #11	PCIe* #12	PCIe* #13	PCIe* #14	PCIe* #15	PCIe* #16	PCIe* #17	PCIe* #18	PCIe* #19	PCIe* #20	PCIe* #21	PCIe* #22	PCIe* #23	PCIe* #24
Intel® RST for PCIe Storage																														

The PCH Flexible I/O Lane support will vary depending on the PCH SKU. For more information, refer to [PCH SKUs](#) on page 16.

The 30 HSIO lanes on PCH-H supports the following configurations:

- Up to 24 PCIe\* Lanes
  - A maximum of 16 PCIe Ports (or devices) can be enabled

- When a GbE Port is enabled, the maximum number of PCIe Ports (or devices) that can be enabled reduces based off the following:
  - Maximum PCIe Ports (or devices) = 16 - GbE (0 or 1)
  - PCIe Lanes 1-4 (PCIe Controller #1), 5-8 (PCIe Controller #2), 9-12 (PCIe Controller #3), 13-16 (PCIe Controller #4), 17-20 (PCIe Controller #5), and 21-24 (PCIe Controller #6) can be individually configured
- 2. Up to 10 SATA Lanes
  - A maximum of 8 SATA Ports (or devices) can be enabled
  - SATA Lane 0 has the flexibility to be mapped to Flex I/O Lane 16 or 18
  - SATA Lane 1 has the flexibility to be mapped to Flex I/O Lane 17 or 19
- 3. Up to 6 USB 3.2 Gen 1x1/2x1 Lanes and 4 USB 3.2 Gen 1x1 Lanes
  - A maximum of 10 USB 3.2 Ports (or devices) can be enabled
  - USB 3.2 Gen 1x1 = 5 GT/s
  - USB 3.2 Gen 2x1 = 10 GT/s
- 4. Up to 4 GbE Lanes
  - A maximum of 1 GbE Port (or device) can be enabled
- 5. Supports Remapped (Intel® Rapid Storage Technology) PCIe Storage Devices
  - Specific PCIe lanes can be used for a x2 or x4 Intel® Rapid Storage Technology Remapped PCIe NVMe SSD or a x2 or x4 Next Generation Intel® Optane™ Memory Device.
  - Not all PCH PCIe lanes support Intel® RST for PCIe Storage Devices. For more information, refer to [PCH SKUs](#) on page 16 for specific PCH SKU PCIe lane support
- 6. For unused SATA/PCIe Combo Lanes, Flex I/O Lanes that can be configured as PCIe or SATA, the lanes must be statically assigned to SATA or PCIe via the SATA/PCIe Combo Port Soft Straps explained in the SPI Programming Guide.

### 3.1.2 Flexible I/O Lane Selection

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

#### PCIe\*/SATA Lane Selection

In addition to static configuration via soft straps, Flexible I/O Lanes that have PCIe\*/SATA multiplexing can be configured via SATAxPCIe signaling to support implementation like SATA Express or mSATA, where the port configuration is selected by the type of the add-in card that is used.

## 4.0 Memory Mapping

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

### 4.1 Functional Description

Topics Covered:

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

#### 4.1.1 PCI Devices and Functions

The PCH incorporates a variety of PCI devices and functions, as shown in the following table. If a particular system platform does not want to support any one of the Device Functions, with the exception of D30:F0, they can individually be disabled. The integrated Gigabit Ethernet controller will be disabled if no Platform LAN Connect component is detected (refer to [Gigabit Ethernet Controller](#) on page 137).

When a function is disabled, it does not appear to the software. A disabled function will not respond to any register reads or writes, insuring that these devices appear hidden to software.

**Table 8. PCI Devices and Functions**

Devices and Function	Description
Bus M: Device 31: Function 0	LPC Interface (eSPI Enable Strap = 0) eSPI Interface (eSPI Enable Strap = 1)
Bus M: Device 31: Function 1	P2SB
Bus M: Device 31: Function 2	PMC
Bus M: Device 31: Function 3	cAVS (Audio, Voice, Speech)
Bus M: Device 31: Function 4	SMBus Controller
Bus M: Device 31: Function 5	SPI
Bus M: Device 31: Function 6	GbE Controller
Bus M: Device 31: Function 7	Intel® Trace Hub
Bus M: Device 30: Function 0	UART #0
Bus M: Device 30: Function 1	UART #1
Bus M: Device 30: Function 2	SPI #0
Bus M: Device 30: Function 3	SPI #1
Bus M: Device 29: Function 0	PCIe* Port 9
<i>continued...</i>	

Devices and Function	Description
Bus M: Device 29: Function 1	PCIe Port 10
Bus M: Device 29: Function 2	PCIe Port 11
Bus M: Device 29: Function 3	PCIe Port 12
Bus M: Device 29: Function 4	PCIe Port 13
Bus M: Device 29: Function 5	PCIe Port 14
Bus M: Device 29: Function 6	PCIe Port 15
Bus M: Device 29: Function 7	PCIe Port 16
Bus M: Device 28: Function 0	PCIe Port 1
Bus M: Device 28: Function 1	PCIe Port 2
Bus M: Device 28: Function 2	PCIe Port 3
Bus M: Device 28: Function 3	PCIe Port 4
Bus M: Device 28: Function 4	PCIe Port 5
Bus M: Device 28: Function 5	PCIe Port 6
Bus M: Device 28: Function 6	PCIe Port 7
Bus M: Device 28: Function 7	PCIe Port 8
Bus M: Device 27: Function 0	PCIe Port 17
Bus M: Device 27: Function 1	PCIe Port 18
Bus M: Device 27: Function 2	PCIe Port 19
Bus M: Device 27: Function 3	PCIe Port 20
Bus M: Device 27: Function 4	PCIe Port 21
Bus M: Device 27: Function 5	PCIe Port 22
Bus M: Device 27: Function 6	PCIe Port 23
Bus M: Device 27: Function 7	PCIe Port 24
Bus M: Device 25: Function 2	UART Controller #2 (Note: ACPI Device)
Bus M: Device 24: Function 0	Reserved. Used by the NVM Remapping.
Bus M: Device 23: Function 0	SATA Controller
Bus M: Device 22: Function 0	Intel® MEI #1
Bus M: Device 22: Function 1	Intel® MEI #2
Bus M: Device 22: Function 2	IDE-Redirection (IDE-R)
Bus M: Device 22: Function 3	Keyboard and Text (KT) Redirection
Bus M: Device 22: Function 4	Intel® MEI #3
Bus M: Device 22: Function 5	Intel® MEI #4
Bus M: Device 22: Function 7	WLAN
Bus M: Device 21: Function 0	I <sup>2</sup> C Controller #0
Bus M: Device 21: Function 1	I <sup>2</sup> C Controller #1
continued...	

Devices and Function	Description
Bus M: Device 21: Function 2	I <sup>2</sup> C Controller #2
Bus M: Device 21: Function 3	I <sup>2</sup> C Controller #3
Bus M: Device 20: Function 0	xHCI Controller
Bus M: Device 20: Function 1	USB Device Controller
Bus M: Device 20: Function 2	Shared SRAM
Bus M: Device 20: Function 3	CNVi: Wi-Fi*
Bus M: Device 20: Function 5	SDXC
Bus M: Device 19: Function 0	Integrated Sensor Hub
Bus M: Device 18: Function 0	Thermal Subsystem
Bus M: Device 18: Function 2	PMT
Bus M: Device 18: Function 6	SPI #2

### 4.1.2 Fixed I/O Address Ranges

Table below shows the Fixed I/O decode ranges from the processor perspective.

#### NOTE

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

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

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

**Table 9. Fixed I/O Ranges Decoded by PCH**

I/O Address	Read Target	Write Target	Internal Unit (Unless[E]: External) <sup>2</sup>	Separate Enable/Disable
20h – 21h	Interrupt Controller	Interrupt Controller	Interrupt	None
24h – 25h	Interrupt Controller	Interrupt Controller	Interrupt	None
28h – 29h	Interrupt Controller	Interrupt Controller	Interrupt	None
2Ch – 2Dh	Interrupt Controller	Interrupt Controller	Interrupt	None
2Eh-2Fh	Super I/O	Super I/O	[E] Forwarded to LPC/eSPI	Yes. IOE.SE
30h – 31h	Interrupt Controller	Interrupt Controller	Interrupt	None
34h – 35h	Interrupt Controller	Interrupt Controller	Interrupt	None
<i>continued...</i>				

I/O Address	Read Target	Write Target	Internal Unit (Unless[E]: External) <sup>2</sup>	Separate Enable/Disable
38h – 39h	Interrupt Controller	Interrupt Controller	Interrupt	None
3Ch – 3Dh	Interrupt Controller	Interrupt Controller	Interrupt	None
40h	Timer/Counter	Timer/Counter	8254 Timer	None
42h-43h	Timer/Counter	Timer/Counter	8254 Timer	None
4Eh-4Fh	Microcontroller	Microcontroller	[E] Forwarded to LPC/eSPI	Yes. IOE.ME2
50h	Timer/Counter	Timer/Counter	8254 Timer	None
52h-53h	Timer/Counter	Timer/Counter	8254 Timer	None
60h	Keyboard Controller	Keyboard Controller	[E] Forwarded to LPC/eSPI	Yes, with 64h. IOE.KE
61h	NMI Controller	NMI Controller	CPU I/F	None
62h	Microcontroller	Microcontroller	[E] Forwarded to LPC/eSPI	Yes, with 66h. IOE.ME1
63h	NMI Controller <sup>1</sup>	NMI Controller <sup>1</sup>	CPU I/F	Yes, alias to 61h. GIC.P61AE
64h	Keyboard Controller	Keyboard Controller	[E] Forwarded to LPC/eSPI	Yes, with 60h. IOE.KE
65h	NMI Controller <sup>1</sup>	NMI Controller <sup>1</sup>	CPU I/F	Yes, alias to 61h. GIC.P61AE
66h	Microcontroller	Microcontroller	[E] Forwarded to LPC/eSPI	Yes, with 62h. IOE.ME1
67h	NMI Controller <sup>1</sup>	NMI Controller <sup>1</sup>	CPU I/F	Yes, alias to 61h. GIC.P61AE
70h	RTC Controller	NMI and RTC Controller	RTC	None
71h	RTC Controller	RTC Controller	RTC	None
72h	RTC Controller	RTC Controller	RTC	None. Alias to 70h if RC.UE=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 <sup>3</sup>	LPC/eSPI or PCIe*	LPC/eSPI or PCIe*	Read: LPC/[E] eSPI or PCIe* Write: [E] LPC/[E] eSPI or	None. PCIe* if GCS.RPR='1', else LPC/eSPI

continued...

<b>I/O Address</b>	<b>Read Target</b>	<b>Write Target</b>	<b>Internal Unit (Unless[E]: External)<sup>2</sup></b>	<b>Separate Enable/Disable</b>
			[E] PCIe*	
84h - 86h	LPC/eSPI or PCIe*	LPC/eSPI or PCIe*	Read: LPC/[E] eSPI or PCIe* Write: [E] LPC/[E] eSPI or [E] PCIe*	None. PCIe* if GCS.RPR='1', else LPC/eSPI
88h	LPC/eSPI or PCIe*	LPC/eSPI or PCIe*	Read: LPC/[E] eSPI or PCIe* Write: [E] LPC/[E] eSPI or [E] PCIe*	None. PCIe* if GCS.RPR='1', else LPC/eSPI
8Ch - 8Eh	LPC/eSPI or PCIe*	LPC/eSPI or PCIe*	Read: LPC/[E] eSPI or PCIe* Write: [E] LPC/[E] eSPI or [E] PCIe*	None. PCIe* if GCS.RPR='1', else LPC/eSPI
90h	LPC/eSPI	LPC/eSPI	Read: LPC/[E] eSPI Write: [E] LPC/[E] eSPI	None. Alias to 80h
92h	Reset Generator	Reset Generator	CPU I/F	None
94h - 96h	LPC/eSPI	LPC/eSPI	Read: LPC/[E] eSPI Write: [E] LPC/[E] eSPI	None. Alias to 8xh
98h	LPC/eSPI	LPC/eSPI	Read: LPC/[E] eSPI Write: [E] LPC/[E] eSPI	None. Alias to 88h
9Ch - 9Eh	LPC/eSPI	LPC/eSPI	Read: LPC/[E] eSPI Write: [E] LPC/[E] eSPI	None. Alias to 8xh
A0h - A1h	Interrupt Controller	Interrupt Controller	Interrupt	None
A4h - A5h	Interrupt Controller	Interrupt Controller	Interrupt	None
A8h - A9h	Interrupt Controller	Interrupt Controller	Interrupt	None
ACH - ADh	Interrupt Controller	Interrupt Controller	Interrupt	None
B0h - B1h	Interrupt Controller	Interrupt Controller	Interrupt	None
B2h - B3h	Power Management	Power Management	Power Management	None
B4h - B5h	Interrupt Controller	Interrupt Controller	Interrupt	None
<b>continued...</b>				

I/O Address	Read Target	Write Target	Internal Unit (Unless[E]: External) <sup>2</sup>	Separate Enable/Disable
B8h - B9h	Interrupt Controller	Interrupt Controller	Interrupt	None
BCh - BDh	Interrupt Controller	Interrupt Controller	Interrupt	None
200h-207h	Gameport Low	Gameport Low	[E] Forwarded to LPC/eSPI	Yes. IOE.LGE
208h-20Fh	Gameport High	Gameport High	[E] Forwarded to LPC/eSPI	Yes. IOE.HGE
4D0h - 4D1h	Interrupt Controller	Interrupt Controller	Interrupt Controller	None
CF9h	Reset Generator	Reset Generator	Interrupt controller	None
Notes: 1. Only if the Port 61 Alias Enable bit (GIC.P61AE) bit is set. Otherwise, the cycle is internally terminated by the PCH. 2. Destination of LPC/eSPI is depending on the eSPI/LPC Pin Strap, '1': eSPI, '0': LPC. 3. This includes byte, word or double-word (DW) access at I/O address 80h.				

### 4.1.3 Variable I/O Decode Ranges

Table below 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 be some unpredictable results if the configuration software allows conflicts to occur. The PCH does not perform any checks for conflicts.

**Table 10. Variable I/O Decode Ranges**

Range Name	Mappable	Size (Bytes)	Target
ACPI	Anywhere in 64K I/O Space	256	Power Management
IDE Bus Master	Anywhere in 64K I/O Space	16 or 32 Bytes	Intel® AMT IDE-R
SMBus	Anywhere in 64K I/O Space	32	SMB Unit
TCO	Anywhere in 64K I/O Space	32	SMB Unit
Parallel Port	3 ranges in 64K I/O Space	8	LPC/eSPI
Serial Port 1	8 Ranges in 64K I/O Space	8	LPC/eSPI
Serial Port 2	8 Ranges in 64K I/O Space	8	LPC/eSPI
Serial Port 3	8 Ranges in 64K I/O space	8	LPC/eSPI
Floppy Disk Controller	2 Ranges in 64K I/O Space	8	LPC/eSPI
LPC Generic 1	Anywhere in 64K I/O Space	4 to 256 Bytes	LPC/eSPI
LPC Generic 2	Anywhere in 64K I/O Space	4 to 256 Bytes	LPC/eSPI
LPC Generic 3	Anywhere in 64K I/O Space	4 to 256 Bytes	LPC/eSPI
<i>continued...</i>			

Range Name	Mappable	Size (Bytes)	Target
LPC Generic 4	Anywhere in 64K I/O Space	4 to 256 Bytes	LPC/eSPI
IO Trapping Ranges	Anywhere in 64K I/O Space	1 to 256 Bytes	Trap
Serial ATA Index/Data Pair	Anywhere in 64K I/O Space	16	SATA Host Controller
PCI Express* Root Ports	Anywhere in 64K I/O Space	I/O Base/Limit	PCI Express* Root Ports 1-6
Keyboard and Text (KT)	Anywhere in 64K I/O Space	8	Intel® AMT Keyboard and Text
<i>Note:</i> All ranges are decoded directly from DMI.			

## 4.2 Memory Map

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

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

### NOTE

Total ports are different for different SKUs.

**Table 11. PCH Memory Decode Ranges (Processor Perspective)**

Memory Range	Target	Dependency/Comments
000E 0000h - 000E FFFFh	LPC/eSPI or SPI	Bit 6 in BIOS Decode Enable Register is set.
000F 0000h - 000F FFFFh	LPC/eSPI or SPI	Bit 7 in BIOS Decode Enable Register is set.
FECX X000h - FECX X040h	I/O(x)APIC inside PCH	XX controlled via APIC Range Select (ASEL) field and APIC Enable (AEN) bit.
FECX X000h - FECX XFFFh	PCIe* port N (N=1 to 24)	X controlled via PCIe* root port N IOxAPIC Range Base/Limit registers and Port N I/OxApic Enable (PAE) is set
FEF0 0000h - FFFF FFFFh	LPC/eSPI or SPI	uCode Patch Region Enable UCPR.UPRE is set
FFC0 0000h - FFC7 FFFFh FF80 0000h - FF87 FFFFh	LPC/eSPI or SPI	Bit 8 in BIOS Decode Enable Register is set
FFC8 0000h - FFCF FFFFh FF88 0000h - FF8F FFFFh	LPC/eSPI or SPI	Bit 9 in BIOS Decode Enable Register is set
FFD0 0000h - FFD7 FFFFh FF90 0000h - FF97 FFFFh	LPC/eSPI or SPI	Bit 10 in BIOS Decode Enable Register is set
FFD8 0000h - FFD7 FFFFh FF98 0000h - FF9F FFFFh	LPC/eSPI or SPI	Bit 11 in BIOS Decode Enable Register is set
FFE0 0000h - FFE7 FFFFh FFA0 0000h - FFA7 FFFFh	LPC/eSPI or SPI	Bit 12 in BIOS Decode Enable Register is set Refer to <a href="#">Table 12</a> on page 36 for swappable ranges
FFE8 0000h - FFEF FFFFh FFA8 0000h - FFAF FFFFh	LPC/eSPI or SPI	Bit 13 in BIOS Decode Enable Register is set
<i>continued...</i>		

Memory Range	Target	Dependency/Comments
		Refer to <a href="#">Table 12</a> on page 36 for swappable ranges
FFF0 0000h - FFF7 FFFFh FFB0 0000h - FFB7 FFFFh	LPC/eSPI or SPI	Bit 14 in BIOS Decode Enable Register is set Refer to <a href="#">Table 12</a> on page 36 for swappable ranges
FFFC 0000h - FFFF FFFFh	LPC/eSPI or SPI	Always enabled. Refer to <a href="#">Table 12</a> on page 36 for swappable ranges
FFF8 0000h - FFFB FFFFh FFB8 0000h - FFBF FFFFh	LPC/eSPI or SPI	Always enabled. Refer to <a href="#">Table 12</a> on page 36 for swappable ranges
FF70 0000h - FF7F FFFFh FF30 0000h - FF3F FFFFh	LPC/eSPI or SPI	Bit 3 in BIOS Decode Enable Register is set
FF60 0000h - FF6F FFFFh FF20 0000h - FF2F FFFFh	LPC/eSPI or SPI	Bit 2 in BIOS Decode Enable Register is set
FF50 0000h - FF5F FFFFh FF10 0000h - FF1F FFFFh	LPC/eSPI or SPI	Bit 1 in BIOS Decode Enable Register is set
FF40 0000h - FF4F FFFFh FF00 0000h - FF0F FFFFh	LPC/eSPI or SPI	Bit 0 in BIOS Decode Enable Register is set
FED0 X000h - FED0 X3FFh	HPET	BIOS determines “fixed” location which is one of four 1 KB ranges where X (in the first column) is 0h, 1h, 2h, or 3h.
FED4 0000h - FED4 7FFFh	LPC or SPI	TPM and Trusted Mobile KBC
FED4 C000h - FED4 FFFFh	PCH Internal (PSF Error Handler)	Always enabled
FED5 0000h - FED5 FFFFh	Intel® CSME	Always enabled
FED7 0000h - FED7 4FFFh	Internal Device	Security feature related
128 KB anywhere in 4 GB range	LAN Controller (CSR registers)	Enable via standard PCI mechanism (Device 31:Function 6)
4 KB anywhere in 4 GB range	LAN Controller (LAN space on Flash)	Enable via standard PCI mechanism (Device 31:Function 6)
64 KB anywhere in 64-bit address range	USB 3.2 Gen 2x1 (10 Gb/s) Host Controller	Enable via standard PCI mechanism (Device 20, Function 0)
2 MB anywhere in 4 GB range	USB 3.2 Gen 1x1 (5 Gb/s) Device Controller	Enable via standard PCI mechanism (Device 20, Function 1)
24 KB anywhere in 4 GB range	USB 3.2 Gen 1x1 (5 Gb/s) 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)
64 KB anywhere in 4 GB range	LPC/eSPI	LPC Generic Memory Range. Enable via setting bit[0] of the LPC Generic Memory Range register (D31:F0:offset 98h).
32 Bytes anywhere in 64-bit address range	SMBus	Enable via standard PCI mechanism (Device 31: Function 4)
2 KB anywhere above 64 KB to 4 GB range	SATA Host Controller	AHCI memory-mapped registers. Enable via standard PCI mechanism (Device 23: Function 0)
<b>continued...</b>		

Memory Range	Target	Dependency/Comments
Memory Base/Limit anywhere in 4 GB range	PCI Express* Root Ports	Enable via standard PCI mechanism
Prefetchable Memory Base/Limit anywhere in 64-bit address range	PCI Express* Root Ports	Enable via standard PCI mechanism
4 KB anywhere in 64-bit address range	Thermal Reporting	Enable via standard PCI mechanism (Device 18: Function 0)
16 Bytes anywhere in 64-bit address range	Intel® MEI #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)
Eight 4 KB slots anywhere in 64-bit address range	UART, GPI and I <sup>2</sup> C controllers	Enable via standard PCI mechanism
4 KB slots anywhere in 64-bit address range	SDXC controllers	Enable via standard PCI mechanism
1 MB (BAR0) or 4 KB (BAR1) in 4 GB 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

### 4.2.1 Boot Block Update Scheme

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

For FWH when top swap is enabled, accesses to FFFF 0000h-FFFF FFFFh are directed to FFFE 0000h-FFFE FFFFh and vice versa. When the Top Swap Enable bit is 0, the PCH will not invert A16.

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

**Table 12. Boot Block Update Scheme**

BOOT_BLOCK_SIZE Value	Accesses to	Being Directed to
000 (64 KB)	FFFF 0000h - FFFF FFFFh	FFFE 000h - FFFE FFFFh and vice versa
001 (128 KB)	FFFE 0000h - FFFF FFFFh	FFFC 0000h - FFFD FFFFh and vice versa
010 (256 KB)	FFFC 0000h - FFFF FFFFh	FFF8 0000h - FFFB FFFFh and vice versa
011 (512 KB)	FFF8 0000h - FFFF FFFFh	FFF0 0000h - FFF7 FFFFh and vice versa
100 (1 MB)	FFF0 0000h - FFFF FFFFh	FFE0 0000h - FFEF FFFFh and vice versa
101 - 111	Reserved	Reserved
<i>Note:</i> This bit is automatically set to 0 by RTCRST#, but not by PLTRST#.		

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

The algorithm is:

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

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

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

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

## 5.0 System Management

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

### Acronyms

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

## 5.1 Features

The following features and functions are supported by the PCH:

- First timer timeout to generate SMI# after programmable time:
  - The first timer timeout causes an SMI#, allowing SMM-based recovery from OS lock up
- 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#

### 5.1.1 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.

#### Handling an Intruder

The PCH has an input signal, INTRUDER#, that can be attached to a switch that is activated by the system's case being open. This input has a two RTC clock debounce.

1. If INTRUDER# goes active (after the debouncer), this will set the INTRD\_DET bit in the TCO2\_STS register. The INTRD\_SEL bits in the TCO\_CNT register can enable the PCH to cause an SMI# or interrupt. The BIOS or interrupt handler can then cause a transition to the S5 state by writing to the SLP\_EN bit.

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

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

#### NOTE

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

### 5.1.2 TCO Modes

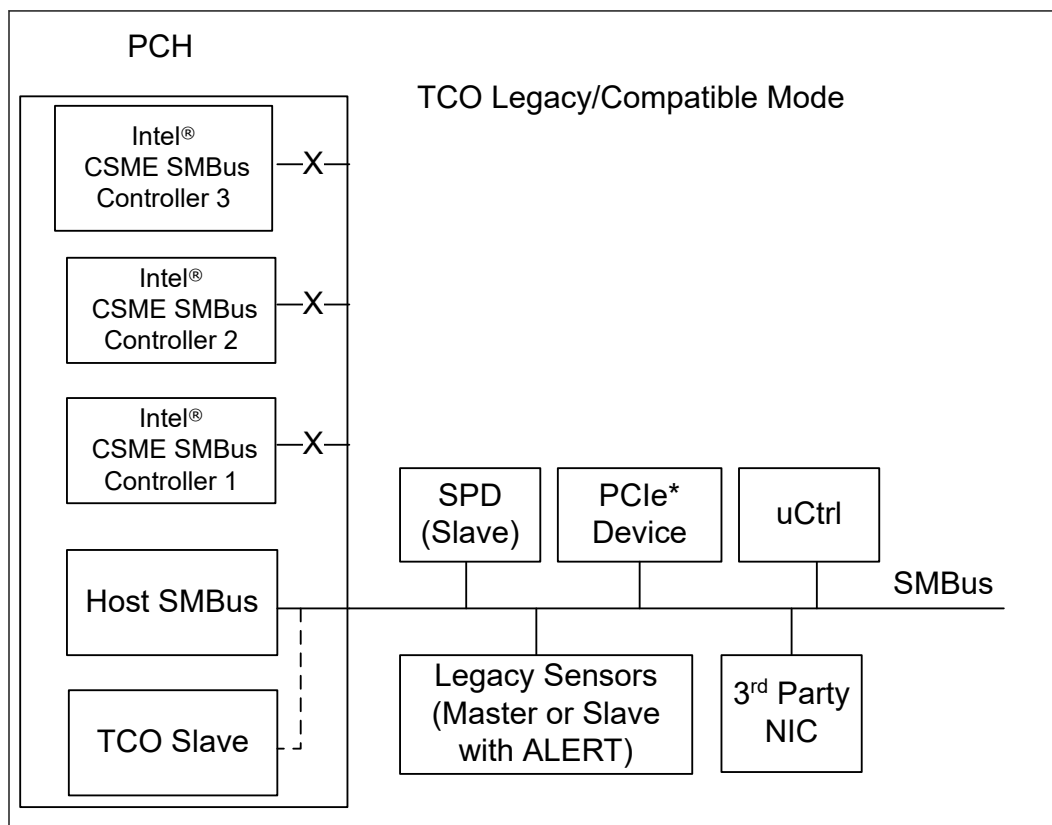
Topics Covered:

- TCO Compatible Mode
- Advanced TCO Mode

#### TCO Compatible Mode

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

**Figure 2. TCO Compatible Mode SMBus Configuration**



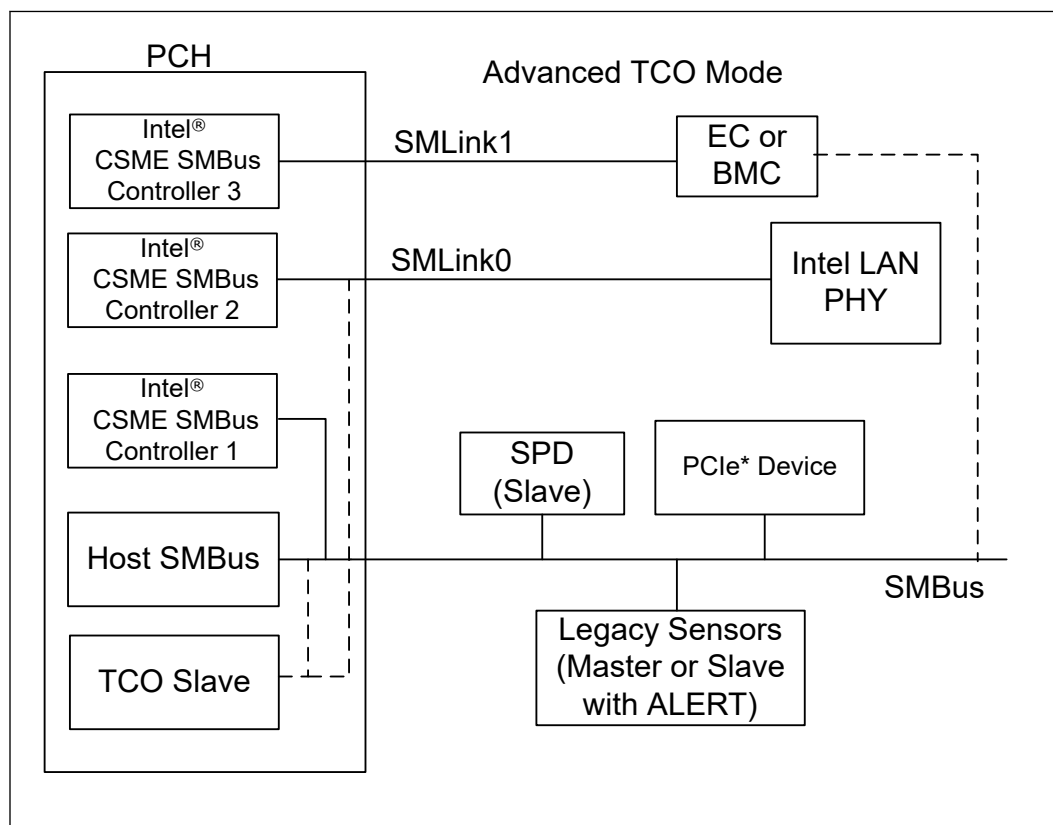
In TCO Legacy/Compatible mode the PCH can function directly with an external LAN controller or equivalent external LAN controller to report messages to a network management console without the aid of the system processor. This is crucial in cases where the processor is malfunctioning or cannot function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

**Table 13. Event Transitions that Cause Messages**

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

### 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 slave can either be connected to the host SMBus or the SMLink0.
- SMLink0 is targeted for integrated LAN use. When an Intel LAN PHY is connected to SMLink0, a soft strap must be set to indicate that the PHY is connected to SMLink0. When the Fast Mode is enabled using a soft strap, the interface will be running at the frequency of up to 1 MHz depending on different factors such as board routing or bus loading.
- SMLink1 can be connected to an Embedded Controller (EC) or Baseboard Management Controller (BMC) use. In the case where a BMC is connected to SMLink1, the BMC communicates with the Intel® Converged Security and Management Engine through the Intel® CSME SMBus connected to SMLink1. The host and TCO slave communicate with BMC through SMBus.

**Figure 3. Advanced TCO Mode**

## 6.0 High Precision Event Timer (HPET)

Specification	Location
IA-PC HPET (High Precision Event Timers) Specification, Revision 1.0a	<a href="http://www.intel.com/content/dam/www/public/us/en/documents/technical-specifications/software-developers-hpet-spec-1-0a.pdf">http://www.intel.com/content/dam/www/public/us/en/documents/technical-specifications/software-developers-hpet-spec-1-0a.pdf</a>

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

The PCH provides eight timers with the following characteristics:

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

Timer 0 supports periodic interrupts.

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

### 6.1 Timer Accuracy

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

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

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

The main counter uses the PCH's 24 MHz XTAL as its clock. The accuracy of the main counter is as accurate as the crystal that is used in the system. The XTAL clock frequency is determined by the pin strap that is sampled on RSMRST#.

## 6.2 Timer Off-load

The PCH supports a timer off-load feature that allows the HPET timers to remain operational during very low power S0 operational modes when the 24 MHz clock is disabled. The clock source during this off-load is the Real Time Clocks 32.768 kHz clock. This clock is calibrated against the 24 MHz clock during boot time to an accuracy that ensures the error introduced by this off-load is less than 10 ppb (0.000001%).

When the 24 MHz clock is active, the 64-bit counter will increment by one each cycle of the 24 MHz clock when enabled. When the 24 MHz clock is disabled, the timer is maintained using the RTC clock. The long-term (> 1 ms) frequency drift allowed by the HPET specification is 500 ppm. The off-load mechanism ensures that it contributes < 1 ppm to this, which will allow this specification to be easily met given the clock crystal accuracies required for other reasons.

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

The HPET timer in the PCH runs typically on the 24 MHz crystal clock and is off-loaded to the 32 kHz clock once the processor enters C10. This is the state where there are no C10 wake events pending and when the off-load calibrator is not running. HPET timer re-uses this 28 bit calibration value calculated by PMC when counting on the 32 kHz clock. During C10 entry, PMC sends an indication to HPET to off-load and keeps the indication active as long as the processor is in C10 on the 32 kHz clock. The HPET counter will be off-loaded to the 32 kHz clock domain to allow the 24 MHz clock to shut down when it has no active comparators.

### Theory of Operation

- The Off-loadable Timer Block consists of a 64b fast clock counter and an 82b slow clock counter. During fast clock mode the counter increments by one on every rising edge of the fast clock. During slow clock mode, the 82-bit slow clock counter will increment by the value provided by the Off-load Calibrator.
- The Off-loadable Timer will accept an input to tell it when to switch to the slow RTC clock mode and provide an indication of when it is using the slow clock mode. The switch will only take place on the slow clock rising edge, so for the 32 kHz RTC clock the maximum delay is around 30  $\mu$ s to switch to or from slow clock mode. Both of these flags will be in the fast clock domain.
- When transitioning from fast clock to slow clock, the fast clock value will be loaded into the upper 64b of the 82b counter, with the 18 LSBs set to zero. The actual transition through happens in two stages to avoid metastability. There is a fast clock sampling of the slow clock through a double flop synchronizer.
- Following a request to transition to the slow clock, the edge of the slow clock is detected and this causes the fast clock value to park. At this point the fast clock can be gated.
- On the next rising edge of the slow clock, the parked fast clock value (in the upper 64b of an 82b value) is added to the value from the Off-load Calibrator. On subsequent edges while in slow clock mode the slow clock counter increments its count by the value from the Off-load Calibrator.
- When transitioning from slow clock to fast clock, the fast clock waits until it samples a rising edge of the slow clock through its synchronizer and then loads the upper 64b of the slow clock value as the fast count value. It then de-asserts the indication that slow clock mode is active. The 32 kHz clock counter no longer counts. The 64-bit MSB will be over-written when the 32 kHz counter is reloaded.

once conditions are met to enable the 32 kHz HPET counter but the 18-bit LSB is retained and it is not cleared out during the next reload cycle to avoid losing the fractional part of the counter.

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

## 6.3 Interrupt Mapping

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

### Mapping Option #1 (Legacy Replacement Option)

**Table 14. Legacy Replacement Routing**

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

### Mapping Option #2 (Standard Option)

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

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

- Timer 0 and 1: IRQ20, 21, 22, and 23 (I/O APIC only).
- Timer 2: IRQ11 (8259 or I/O APIC) and IRQ20, 21, 22, and 23 (I/O APIC only).
- Timer 3: IRQ12 (8259 or I/O APIC) and IRQ 20, 21, 22, and 23 (I/O APIC only).

---

**NOTES**

1. Interrupts from Timer 4, 5, 6, 7 can only be delivered via direct FSB interrupt messages.
  2. System architecture changes since the HPET specification 1.0 was released have made some of the terminology used obsolete. In particular the reference to a Front Side Bus (FSB) has no relevance to current platforms, as this interface is no longer in use. For consistency with the HPET specification though, the FSB and specifically the FSB Interrupt Delivery terminology has been maintained. Where the specification refers to FSB, this should be read as 'processor message interface'; independent of the physical attach mechanism.
- 

**Mapping Option #3 (Processor Message Option)**

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

When the interrupt is delivered to the processor, the message is delivered to the address indicated in the Tn\_PROCMMSG\_INT\_ADDR field. The data value for the write cycle is specified in the Tn\_PROCMMSG\_INT\_VAL field.

---

**NOTE**

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

---

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

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

## 6.4 Periodic Versus Non-Periodic Modes

Topics covered here are, Non-Periodic Mode and Periodic Mode.

**Non-Periodic Mode**

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

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

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

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

1. Set TIMER0\_VAL\_SET\_CNF bit

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

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

---

### WARNING

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

---

All of the timers support non-periodic mode.

### Periodic Mode

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

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

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

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

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

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

Software sets the `ENABLE_CNF` bit to enable interrupts.

---

**NOTE**

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

---

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

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

## 6.5 Enabling the Timers

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

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

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

## 6.6 Interrupt Levels

Interrupts directed to the internal 8259s are active high. Refer to [Advanced Programmable Interrupt Controller \(APIC\) \(D31:F0\)](#) on page 153 for information regarding the polarity programming of the I/O APIC for detecting internal interrupts.

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

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

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

## 7.0 Thermal Management

---

### 7.1 PCH Thermal Sensor

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

#### 7.1.1 Modes of Operation

The DTS has two usages when enabled:

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

There are two thermal alert capabilities:

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

#### 7.1.2 Temperature Trip Point

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

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

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

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

#### 7.1.3 Thermal Sensor Accuracy $T_{\text{accuracy}}$

The PCH thermal sensor accuracy is:

- $\pm 5$  °C over the temperature range from 50 °C to 110 °C.
- $\pm 7$  °C over the temperature range from 30 °C to 50 °C.

- $\pm 10$  °C over the temperature range from -10 °C to 30 °C.

#### 7.1.4 Thermal Reporting to an EC

To support a platform EC that is managing the system thermals, the PCH provides the ability for the EC to read the PCH temperature over SMLink1 or over eSPI interface.

The PCH supports temperature reporting over SMLink1 using the General Purpose Block Read (GPBR). To enable the support, the related soft straps, SMLink1 GP Target Address and SMLink1 GP Target Address Enabled, must be set appropriately. Refer to the Platform SPI Programming Guide for more information on the soft straps. The EC needs to issue an SMBus Block Read Command (40h) and receives a single byte of data indicating the PCH temperature. Upon reset, the value driven to the EC will be FFh. This indicates that BIOS has not enabled the sensor yet and the EC can assume that the system is in the boot phase with unknown temperature.

After the sensor is enabled, the EC will receive a value between 0h and 7Fh (0 °C to 127 °C). If the EC ever sees a value between 80h and FEh, that indicates an error has occurred, since the PCH should have shut down the platform before the temperature ever reached 128 °C (Catastrophic trip point will be below 128 °C). The PCH itself does not monitor the temperature and will not flag any error on the temperature value. The polling rate is recommended no more than once every 200 ms, although EC can read as often as it wishes.

---

#### NOTE

The Catastrophic trip value is a 8 bit fuse and is not programmed or accessible by BIOS. It is set to 120 °C.

---

The PCH also supports temperature reporting over eSPI using eSPI OOB channel. For getting the PCH temperature over eSPI, the EC can send a request packet via OOB channel and receive a response packet consisting of 1 byte of PCH temperature. Refer to the [enhanced Serial Peripheral Interface \(eSPI\)](#) on page 117 for detailed information on eSPI request and response packets for getting PCH temperature data.

#### 7.1.5 Thermal Trip Signal (PCHHOT#)

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

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

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

## 8.0 Power and Ground Signals

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

**Table 15. Power Rails Descriptions PCH-H**

Name	Description
VCCA_BCLK_1P05	Analog supply for BCLK circuitries: 1.05 V
VCCA_SRC_1P05	Analog supply for PCIe* clock circuitries: 1.05 V
VCCA_XTAL_1P05	Analog supply for XTAL circuitries: 1.05 V
VCCDUSB_1P05	Supply for USB digital logic: 1.05 V
VCCAPLL_1P05	Analog supply for BCLK/DMI/Audio PLLs: 1.05 V. This rail can be derived from the VCCPRIM_1P05 rail with the proper isolation
VCCPRIM_1P05	<b>Primary Well:</b> 1.05 V. For PCIe*/DMI/USB 3.2/SATA MPHY logic, I/O blocks, SRAM, JTAG, CNVi
VCCDSW_1P05	<b>Deep Sx Well:</b> 1.05 V. This rail is generated by on die DSW low dropout (LDO) linear regulator to supply DSW core logic. When the primary well is powered, under PMC control, the regulation in the DSW LDO is turned off and the bypass option is used to derive DSW 1.05 V supply from primary well 1.05 V supply
VCCPRIM_MPHY_1P05	<b>Mod PHY Primary:</b> 1.05 V. Primary supply for PCIe*/USB 3.2/SATA MPHY logic and PCIe*/USB PLL dividers
VCCAMPHYPLL_1P05	Analog supply for USB 3.2, PCIe* Gen 2/Gen 3, and SATA 6 Gb/s PLLs: 1.05 V
VCCPRIM_1P8	<b>1.8 V Primary Well</b>
VCCPRIM_3P3	<b>3.3 V Primary Well</b>
VCCSPI	<b>SPI Primary Well 3.3 V or 1.8 V</b> , for SPI interface
VCCHDA	<b>3.3 V or 1.8 V Well</b> for Intel® High Definition Audio
VCCDSW_3P3	<b>3.3 V Deep Sx Well</b>
VCCRTC	<p><b>RTC Well Supply.</b> 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.</p> <p><i>Notes:</i> 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.</p> <p>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.</p>
DCPRTC	RTC decoupling capacitor only. This rail should NOT be driven
VCCDPHY_1P24	<b>1.24 V for CNVi logic.</b> This rail is generated internally with a LDO and needs to be routed to the motherboard so that the rail can be supplied back to the SoC
VCCPHVLDO_1P8	<b>1.8 V Primary Well.</b> On the motherboard, this power pin must be connected to VCCPRIM_1P8 rail in Internal 1.8 V VRM Mode and left as no-connect in External 1.8V VRM Mode
VCCPGPPA	<b>1.8 V or 3.3 V for GPP_A group</b>
VCCPGPPBC	<b>1.8 V or 3.3 V for GPP_B and GPP_C groups</b>
continued...	

Name	Description
VCCPGPPD	1.8 V or 3.3 V for GPP_D group
VCCPGPPEF	1.8 V or 3.3 V for GPP_E and GPP_F groups
VCCPGPPG_3P3	3.3 V for GPP_G group
VCCPGPPHK	1.8 V or 3.3 V for GPP_H and GPP_K groups
VCCMPHY_SENSE	1.05 V Sense Line. This power pin should be left as no-connect
VSSMPHY_SENSE	0 V (Ground) Sense Line. This power pin should be left as no-connect
VSS	Ground

## 9.0 Electrical and Thermal Characteristics

This chapter contains the DC characteristics for the PCH.

### 9.1 Power Rail Absolute Maximum Ratings

**Table 16. PCH Absolute Power Rail Maximum and Minimum Ratings**

Voltage Rail	Minimum Limit	Maximum Limits
1.05 V	-0.5 V	1.4 V
1.8 V	-0.5 V	2.3 V
3.3 V	-0.7 V	3.7 V

Above table specifies absolute maximum and minimum ratings. At conditions outside functional operation condition limits, but within absolute maximum and minimum ratings, neither functionality nor long-term reliability can be expected. If a device is returned to conditions within functional operation limits after having been subjected to conditions outside these limits (but within the absolute maximum and minimum ratings) the device may be functional, but with its lifetime degraded depending on exposure to conditions exceeding the functional operation condition limits.

At conditions exceeding absolute maximum and minimum ratings, neither functionality nor long-term reliability can be expected. Moreover, if a device is subjected to these conditions for any length of time, it will likely either not function or its reliability will be severely degraded when returned to conditions within the functional operating condition limits.

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

### 9.2 Thermal Specification

**Table 17. Operating Junction Temperature Range**

Parameter	Minimum	Maximum	Unit
Operating Junction Temperature Range	0	110	°C

## 9.3 General DC Characteristics

**Table 18. PCH Icc<sup>3</sup> with Integrated 1.8 V VRM Mode OFF (Desktop SKUs)**

Voltage Rail	Voltage (V)	S0 Iccmax Current <sup>2</sup> (A)	Sx Icc Idle Current <sup>4</sup> (mA)	S0ix Iccmax Current (mA)	Deep Sx Icc Idle Current (mA)	G3 (μA)
VCCAPLL_1P05	1.05	0.041	0.2	0.801	0	0
VCCA_BCLK_1P05	1.05	0.009	0.1	0.321	0	0
VCCA_SRC_1P05	1.05	0.169	0.313	3.092	0	0
VCCA_XTAL_1P05	1.05	0.005	0.551	0.195	0	0
VCCAMPHYPLL_1P05	1.05	0.145	0.4	4.344	0	0
VCCPRIM_1P05	1.05	7.169+ HSIO Lane Adder refer to <a href="#">Table 22</a> on page 57 column DT	121.697	1.622 A	0	0
VCCPRIM_MPHY_1P05	1.05	0.106	0.307	4.313	0	0
VCCDSW_1P05	1.05	0.01	0.2	0.003	0.2	0
VCCDUSB_1P05	1.05	0.421	5.698	60.397	0	0
VCCHDA	3.3	0.008	0.1	4.998	0	0
	1.8	0.005	0.06	3	0	0
	1.5	0.004	0.05	2.5	0	0
VCCDSW_3P3	3.3	0.113	0.534	0.705	1.05	0
VCCPRIM_3P3	3.3	0.383	0.391	3.378	0	0
VCCPGPPA	3.3	0.102	0.135	0.378	0	0
	1.8	0.061	0.121	0.340	0	0
VCCPGPPBC	3.3	0.344	0.262	0.853	0	0
	1.8	0.206	0.236	0.767	0	0
VCCPGPPD	3.3	0.14	0.122	0.402	0	0
	1.8	0.084	0.110	0.362	0	0
VCCPGPPEF	3.3	0.175	0.2	0.345	0	0
	1.8	0.104	0.180	0.310	0	0
VCCPGPPG_3P3	3.3	0.145	0.1	0.072	0	0
VCCPGPPHK	3.3	0.263	0.2	0.509	0	0
	1.8	0.157	0.180	0.458	0	0
VCCPRIM_1P8	1.8	0.183+ CNVi Adder refer to <a href="#">Table 23</a> on page 57, column DT	6.835	18.202	0	0
VCCRTC <sup>1</sup>	3.0	0.60 mA	0.317	0.075	0.316	6

*continued...*

Voltage Rail	Voltage (V)	S0 Iccmax Current <sup>2</sup> (A)	Sx Icc Idle Current <sup>4</sup> (mA)	S0ix Iccmax Current (mA)	Deep Sx Icc Idle Current (mA)	G3 (μA)
VCCSPI	3.3	0.05	0.1	0.153	0	0
	1.8	0.03	0.09	0.138	0	0

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

**Table 19. PCH Icc<sup>3</sup> with Integrated 1.8 V VRM Mode OFF (H Mobile SKUs)**

Voltage Rail	Voltage (V)	S0 Iccmax Current <sup>2</sup> (A)	Sx Icc Idle Current <sup>4</sup> (mA)	S0ix Iccmax Current (mA)	Deep Sx Icc Idle Current (mA)	G3 (μA)
VCCAPLL_1P05	1.05	0.034	0.2	0.801	0	0
VCCA_BCLK_1P05	1.05	0.007	0.1	0.087	0	0
VCCA_SRC_1P05	1.05	0.141	0.3	0.838	0	0
VCCA_XTAL_1P05	1.05	0.005	0.544	0.195	0	0
VCCAMPHYPLL_1P05	1.05	0.114	0.4	1.192	0	0
VCCPRIM_1P05	1.05	4.174+ HSIO Lane Adder refer to <a href="#">Table 22</a> on page 57 column HALO	40.344	0.477 A	0	0
VCCPRIM_MPHY_1P05	1.05	0.088	0.2	1.22	0	0
VCCDSW_1P05	1.05	0.01	0.2	0.001	0.2	0
VCCDUSB_1P05	1.05	0.33	1.288	16.373	0	0
VCCHDA	3.3	0.006	0.1	4.908	0	0
	1.8	0.004	0.06	2.944	0	0
	1.5	0.003	0.05	2.454	0	0
VCCDSW_3P3	3.3	0.094	0.2	0.705	1.05	0
VCCPRIM_3P3	3.3	0.318	0.3	0.916	0	0
VCCPGPPA	3.3	0.085	0.1	0.103	0	0
	1.8	0.051	0.09	0.092	0	0
VCCPGPPBC	3.3	0.286	0.2	0.232	0	0
	1.8	0.171	0.18	0.208	0	0
VCCPGPPD	3.3	0.117	0.1	0.109	0	0
	1.8	0.070	0.09	0.098	0	0
VCCPGPPEF	3.3	0.145	0.2	0.094	0	0
	1.8	0.087	0.18	0.084	0	0
VCCPGPPG_3P3	3.3	0.121	0.1	0.072	0	0

*continued...*

Voltage Rail	Voltage (V)	S0 Iccmax Current <sup>2</sup> (A)	Sx Icc Idle Current <sup>4</sup> (mA)	S0ix Iccmax Current (mA)	Deep Sx Icc Idle Current (mA)	G3 (μA)
VCCPGPPHK	3.3	0.219	0.2	0.138	0	0
	1.8	0.131	0.18	0.124	0	0
VCCPRIM_1P8	1.8	0.152+ CNVi Adder refer to <a href="#">Table 23</a> on page 57, column HALO	6.607	9.411	0	0
VCCRTC <sup>1</sup>	3.0	0.31 mA	0.299	0.075	0.316	6
VCCSPI	3.3	0.042	0.1	0.153	0	0
	1.8	0.025	0.09	0.138	0	0

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

**Table 20. PCH Icc<sup>3</sup> with Integrated 1.8 V VRM Mode ON (Desktop SKUs)**

Voltage Rail	Voltage (V)	S0 Iccmax Current <sup>2</sup> (A)	Sx Icc Idle Current <sup>4</sup> (mA)	S0ix Iccmax Current (mA)	Deep Sx Icc Idle Current (mA)	G3 (μA)
VCCAPLL_1P05	1.05	0.041	0.2	0.801	0	0
VCCA_BCLK_1P05	1.05	0.009	0.1	0.321	0	0
VCCA_SRC_1P05	1.05	0.169	0.313	3.092	0	0
VCCA_XTAL_1P05	1.05	0.005	0.551	0.195	0	0
VCCAMPHYPLL_1P05	1.05	0.145	0.4	4.344	0	0
VCCPRIM_1P05	1.05	7.169+ HSIO Lane Adder refer to <a href="#">Table 22</a> on page 57, column DT	121.697	1.622 A	0	0
VCCPRIM_MPHY_1P05	1.05	0.106	0.307	4.313	0	0
VCCDSW_1P05	1.05	0.01	0.2	0.003	0.2	0
VCCDUSB_1P05	1.05	0.421	5.698	60.397	0	0
VCCHDA	3.3	0.008	0.1	4.998	0	0
	1.8	0.005	0.06	3	0	0
	1.5	0.004	0.05	2.5	0	0
VCCDSW_3P3	3.3	0.113	0.534	0.705	1.05	0
VCCPRIM_3P3	3.3	0.761+ CNVi Adder refer to <a href="#">Table 24</a> on page 58, column DT	9.451	30.811	0	0

*continued...*

Voltage Rail	Voltage (V)	S0 Iccmax Current <sup>2</sup> (A)	Sx Icc Idle Current <sup>4</sup> (mA)	S0ix Iccmax Current (mA)	Deep Sx Icc Idle Current (mA)	G3 (μA)
VCCPGPPA	3.3	0.102	0.135	0.378	0	0
	1.8	0.061	0.121	0.340	0	0
VCCPGPPBC	3.3	0.344	0.262	0.853	0	0
	1.8	0.206	0.236	0.767	0	0
VCCPGPPD	3.3	0.14	0.122	0.402	0	0
	1.8	0.084	0.110	0.362	0	0
VCCPGPPEF	3.3	0.175	0.2	0.345	0	0
	1.8	0.104	0.18	0.310	0	0
VCCPGPPG_3P3	3.3	0.145	0.1	0.072	0	0
VCCPGPPHK	3.3	0.263	0.2	0.509	0	0
	1.8	0.157	0.18	0.458	0	0
VCCRTC	3.0	0.60 mA	0.317	0.075	0.316	6
VCCSPI	3.3	0.05	0.1	0.153	0	0
	1.8	0.03	0.09	0.138	0	0
<b>Notes:</b> 1. The VCC rail ICC data is taken at 3.0 V while the system is in a mechanical off (G3) state at room temperature. 2. Iccmax Specifications assumes 110 °C. 3. The Iccmax value is a steady state current that can happen after respective power ok has asserted (or reset signal has de-asserted). 4. Sx Icc Idle assumes PCH is idle and Intel® CSME is power gated.						

**Table 21. PCH Icc3 with Integrated 1.8 V VRM Mode ON (H Mobile SKUs)**

Voltage Rail	Voltage (V)	S0 Iccmax Current <sup>2</sup> (A)	Sx Icc Idle Current <sup>4</sup> (mA)	S0ix Iccmax Current (mA)	Deep Sx Icc Idle Current (mA)	G3 (μA)
VCCAPLL_1P05	1.05	0.034	0.2	0.801	0	0
VCCA_BCLK_1P05	1.05	0.007	0.1	0.087	0	0
VCCA_SRC_1P05	1.05	0.141	0.3	0.838	0	0
VCCA_XTAL_1P05	1.05	0.005	0.544	0.195	0	0
VCCAMPHYPLL_1P05	1.05	0.114	0.4	1.192	0	0
VCCPRIM_1P05	1.05	4.174+ HSIO Lane Adder refer to <a href="#">Table 22</a> on page 57, column HALO	40.344	0.477 A	0	0
VCCPRIM_MPHY_1P05	1.05	0.088	0.2	1.22	0	0
VCCDSW_1P05	1.05	0.01	0.2	0.001	0.2	0
VCCDUSB_1P05	1.05	0.33	1.288	16.373	0	0
VCCHDA	3.3	0.007	0.1	4.908	0	0
<b>continued...</b>						

Voltage Rail	Voltage (V)	S0 Iccmax Current <sup>2</sup> (A)	Sx Icc Idle Current <sup>4</sup> (mA)	S0ix Iccmax Current (mA)	Deep Sx Icc Idle Current (mA)	G3 (μA)
VCCDSW_3P3	3.3	0.094	0.2	0.705	1.05	0
VCCPRIM_3P3	3.3	0.536+ CNVi Adder refer to Table 24 on page 58, column HALO	8.732	17.105	0	0
VCCPGPPA	3.3	0.085	0.1	0.103	0	0
VCCPGPPBC	3.3	0.286	0.2	0.232	0	0
VCCPGPPD	3.3	0.117	0.1	0.109	0	0
VCCPGPPEF	3.3	0.145	0.2	0.094	0	0
VCCPGPPG_3P3	3.3	0.121	0.1	0.072	0	0
VCCPGPPHK	3.3	0.219	0.2	0.138	0	0
VCCRTC	3.0	0.31 mA	0.299	0.075	0.316	6
VCCSPI	3.3	0.042	0.1	0.153	0	0
<b>Notes:</b> 1. The VCC rail ICC data is taken at 3.0 V while the system is in a mechanical off (G3) state at room temperature. 2. Iccmax Specifications assumes 110 °C. 3. The Iccmax value is a steady state current that can happen after respective power ok has asserted (or reset signal has de-asserted). 4. Sx Icc Idle assumes PCH is idle and Intel® CSME is power gated. 5. Sx Icc at 3.3 V level is assumed. Sx Icc data at the 1.8 V and/or 1.5 V level not measured.						

**Table 22. PCH VCCPRIM\_1P05 Adder per HSIO Lane**

Details	Icc (HALO) (mA)	Icc (DT) (mA)
Baseline (x4 DMI Gen3)	639	1116
Baseline (x4 DMI Gen2)	607	1084
Each PCIe Gen3 Lane	148	178
Each PCIe Gen2 Lane	123	152
Each USB 3.2 Gen 2x1 (10 Gb/s) Port	182	221
Each USB 3.2 Gen 1x1 (5 Gb/s) Port	125	155
Integrated GbE Port	62	98
Each SATA 6 Gb/s Port	126	167

**Table 23. PCH VCCPRIM\_1P8 Adder with and without CNVi**

Details	Icc (HALO) (mA)	Icc (DT) (mA)
Without CNVi	0	0
With CNVi	582	589

**Table 24. PCH VCCPRIM\_3P3 Adder with and without CNVi**

Details	I <sub>cc</sub> (HALO) (mA)	I <sub>cc</sub> (DT) (mA)
Without CNVi	0	0
With CNVi	582	589

**Table 25. Single-Ended Signal DC Characteristics as Inputs or Outputs**

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
<b>Note:</b> For GPIO pads (GPP) listed in the Associated Signals below, all functions that are multiplexed on GPIO pads will have the same DC characteristics as the GPIO pads. Refer to the <a href="#">General Purpose Input and Output (GPIO)</a> on page 127 for the multiplexed functions on a specific GPIO pad.							
Associated Signals <sup>1</sup> : GPP_A0/RCIN#/ESPI_ALERT1#, GPP_A1/LAD0/ESPI_IO0, GPP_A2/LAD1/ESPI_IO1, GPP_A3/LAD2/ESPI_IO2, GPP_A4/LAD3/ESPI_IO3, GPP_A5/LFRAME#/ESPI_CS0#, GPP_A6/SERIRQ/ESPI_CS1#, GPP_A7/PIRQA#/ESPI_ALERT0#, GPP_A8/CLKRUN#, GPP_A9/CLKOUT_LPC0/ESPI_CLK, GPP_A10/CLKOUT_LPC1, GPP_A11/PME#/SD_VDD2_PWR_EN#, GPP_A12/BM_BUSY#/ISH_GP6/SX_EXIT_HOLDOFF#, GPP_A13/SUSWARN#/SUSPWRDNACK, GPP_A14/SUS_STAT#/ESPI_RESET#, GPP_A15/SUSACK#, GPP_A16/CLKOUT_48, GPP_A17/SD_VDD1_PWR_EN#/ISH_GP7, GPP_A18/ISH_GP0, GPP_A19/ISH_GP1, GPP_A20/ISH_GP2, GPP_A21/ISH_GP3, GPP_A22/ISH_GP4, GPP_A23/ISH_GP5, GPP_D17/DMIC_CLK1/SNDW3_CLK, GPP_D18/DMIC_DATA1/SNDW3_DATA, GPP_D19/DMIC_CLK0/SNDW4_CLK, GPP_D20/DMIC_DATA0/SNDW4_DATA, HDA_BCLK/I2S0_SCLK, HDA_RST#/I2S1_SCLK, HDA_SYNC/I2S0_SFRM, HDA_SDO/I2S0_TXD, HDA_SDI0/I2S0_RXD, HDA_SDI1/I2S1_RXD							
<b>3.3 V Operation</b>							
Input	V <sub>IH</sub>	Input High Voltage Threshold	0.7 x VCC	—	V		
	V <sub>IL</sub>	Input Low Voltage Threshold	—	0.3 x VCC	V		
	I <sub>IL</sub>	Input Leakage Current	-12	12	μA		
	C <sub>IN</sub>	Input Pin Capacitance	—	13	pF		
Output	V <sub>OH</sub>	Output High Voltage Threshold	VCC - 0.45	VCC	V	I <sub>OH</sub> =3 mA	
	V <sub>OL</sub>	Output Low Voltage Threshold	—	0.45	V	I <sub>OL</sub> = -3 mA	
	R <sub>pu</sub>	WPU (WeakPull-Up) Resistance	5K-70% 20K-25%	5K+70% 20K+25%	Ohm		
	R <sub>pd</sub>	WPD (Weak Pull-Down) Resistance	5K-70% 20K-25%	5K+70% 20K+25%	Ohm		
<b>1.8 V Operation</b>							
Input	V <sub>IH</sub>	Input High Voltage Threshold	0.7 x VCC	—	V		
	V <sub>IL</sub>	Input Low Voltage Threshold	—	0.3 x VCC	V		
	I <sub>IL</sub>	Input Leakage Current	-12	12	μA		
	C <sub>IN</sub>	Input Pin Capacitance	—	13	pF		
Output	V <sub>OH</sub>	Output High Voltage Threshold	VCC - 0.45	VCC	V	I <sub>OH</sub> =3 mA	
	V <sub>OL</sub>	Output Low Voltage Threshold	—	0.45	V	I <sub>OL</sub> = -3 mA	
	R <sub>pu</sub>	WPU Resistance	5K-70% 20K-25%	5K+70% 20K+25%	Ohm		
<i>continued...</i>							

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
	$R_{pd}$	WPD Resistance	5K-70% 20K-25%	5K+70% 20K+25%	Ohm		

Note: 1. For GPIO supported voltages, refer to [General Purpose Input and Output \(GPIO\)](#) on page 127.

Note: For GPIO pads (GPP) listed in the Associated Signals below, all functions that are multiplexed on GPIO pads will have the same DC characteristics as the GPIO pads. Refer to [General Purpose Input and Output \(GPIO\)](#) on page 127 for the muxed functions on a specific GPIO pad.

Associated Signals<sup>1</sup>: GPP\_B0/GSPI0\_CS1#, GPP\_B1/GSPI1\_CS1#/TIME\_SYNC1, GPP\_B2/VREALERT#, GPP\_B3/CPU\_GP2, GPP\_B4/CPU\_GP3, GPP\_B5/SRCCLKREQ0#, GPP\_B6/SRCCLKREQ1#, GPP\_B7/SRCCLKREQ2#, GPP\_B8/SRCCLKREQ3#, GPP\_B9/SRCCLKREQ4#, GPP\_B10/SRCCLKREQ5#, GPP\_B11/I2S\_MCLK, GPP\_B12/SLP\_S0#, GPP\_B13/PLTRST#, GPP\_B14/SPKR, GPP\_B15/GSPI0\_CS0#, GPP\_B16/GSPI0\_CLK, GPP\_B17/GSPI0\_MISO, GPP\_B18/GSPI0\_MOSI, GPP\_B19/GSPI1\_CS0#, GPP\_B20/GSPI1\_CLK, GPP\_B21/GSPI1\_MISO, GPP\_B22/GSPI1\_MOSI, GPP\_B23/SML1ALERT#/PCHHOT#, GPP\_C0/SMBCLK, GPP\_C1/SMBDATA, GPP\_C2/SMBALERT#, GPP\_C3/SML0CLK, GPP\_C4/SML0DATA, GPP\_C5/SML0ALERT#, GPP\_C6/SML1CLK, GPP\_C7/SML1DATA, GPP\_C8/UART0A\_RXD, GPP\_C9/UART0A\_TXD, GPP\_C10/UART0A\_RTS#, GPP\_C11/UART0A\_CTS#, GPP\_C12/UART1\_RXD/ISH\_UART1\_RXD, GPP\_C13/UART1\_TXD/ISH\_UART1\_TXD, GPP\_C14/UART1\_RTS#/ISH\_UART1\_RTS#, GPP\_C15/UART1\_CTS#/ISH\_UART1\_CTS#, GPP\_C16/I2C0\_SDA, GPP\_C17/I2C0\_SCL, GPP\_C18/I2C1\_SDA, GPP\_C19/I2C1\_SCL, GPP\_C20/UART2\_RXD, GPP\_C21/UART2\_TXD, GPP\_C22/UART2\_RTS#, GPP\_C23/UART2\_CTS#, GPP\_D0/SPI1\_CS#/SBK0/BK0, GPP\_D1/SPI1\_CLK/SBK1/BK1, GPP\_D2/SPI1\_MISO/SBK2/BK2, GPP\_D3/SPI1\_MOSI/SBK3/BK3, GPP\_D4/ISH\_I2C2\_SDA/I2C3\_SDA/SBK4/BK4, GPP\_D5/I2S2\_SFRM/CNV\_RF\_RESET#, GPP\_D6/I2S2\_TXD/MODEM\_CLKREQ, GPP\_D7/I2S2\_RXD, GPP\_D8/I2S2\_SCL, GPP\_D9/ISH\_SPI\_CS#/GSP12\_CS0#, GPP\_D10/ISH\_SPI\_CLK/GSP12\_CLK, GPP\_D11/ISH\_SPI\_MISO/GP\_BSSB\_CLK/GSP12\_MISO, GPP\_D12/ISH\_SPI\_MOSI/GP\_BSSB\_DI/GSP12\_MOSI, GPP\_D13/ISH\_UART0\_RXD/I2C2\_SDA, GPP\_D14/ISH\_UART0\_TXD/I2C2\_SCL, GPP\_D15/ISH\_UART0\_RTS#/GSP12\_CS1#/CNV\_WFEN, GPP\_D16/ISH\_UART0\_CTS#/CNV\_WCEN, GPP\_D21/SPI1\_IO2, GPP\_D22/SPI1\_IO3, GPP\_D23/ISH\_I2C2\_SCL/I2C3\_SCL, GPD0/BATLOW#, GPD1/ACPRESENT, GPD2/LAN\_WAKE#, GPD3/PWRBTN#, GPD4/SLP\_S3#, GPD5/SLP\_S4#, GPD6/SLP\_A#, GPD7, GPD8/SUSCLK, GPD9/SLP\_WLAN#, GPD10/SLP\_S5#, GPD11/LANPHYPC, SLP\_LAN#, SLP\_SUS#, GPD2/LAN\_WAKE#, DRAM\_RESET#

### 3.3 V Operation

Input	VIH	Input High Voltage Threshold	0.75 x VCC	—	V		
	VIL	Input Low Voltage Threshold	—	0.25 x VCC	V		
	IIL	Input Leakage Current	-10	10	μA		
	CIN	Input Pin Capacitance	—	14	pF		
Output	VOH	Output High Voltage Threshold	VCC - 0.45	VCC	V	I <sub>OH</sub> =3 mA	
	VOL	Output Low Voltage Threshold	—	0.45	V	I <sub>OL</sub> =-3 mA	
	$R_{pu}$	WPU Resistance	1K-50% 5K-70% 20K-25%	1K+100% 5K+70% 20K+25%	Ohm		
	$R_{pd}$	WPD Resistance	5K-70% 20K-35%	5K+70% 20K+35%	Ohm		

### 1.8 V Operation

Input	VIH	Input High Voltage Threshold	0.75 x VCC	—	V		
	VIL	Input Low Voltage Threshold	—	0.25 x VCC	V		
	IIL	Input Leakage Current	-10	10	μA		
	CIN	Input Pin Capacitance	—	14	pF		
Output	V <sub>OH</sub>	Output High Voltage Threshold	VCC - 0.45		V	I <sub>OH</sub> =3 mA	

continued...

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
	V <sub>OL</sub>	Output Low Voltage Threshold	—	0.45	V	I <sub>OL</sub> = -3 mA	
	R <sub>pu</sub>	WPU Resistance	1K-50% 5K-70% 20K-25%	1K+100% 5K+70% 20K+25%	Ohm		
	R <sub>pd</sub>	WPD Resistance	5K-70% 20K-25%	5K+70% 20K+25%	Ohm		

Note: 1. For GPIO supported voltages, refer to [General Purpose Input and Output \(GPIO\)](#) on page 127.

Note: For GPIO pads (GPP) listed in the Associated Signals below, all functions that are multiplexed on GPIO pads will have the same DC characteristics as the GPIO pads. Refer to [General Purpose Input and Output \(GPIO\)](#) on page 127 for the muxed functions on a specific GPIO pad.

Associated Signals<sup>1</sup>: GPP\_G0/SD\_CMD, GPP\_G1/SD\_DATA0, GPP\_G2/SD\_DATA1, GPP\_G3/SD\_DATA2, GPP\_G4/SD\_DATA3, GPP\_G5/SD\_CD#, GPP\_G6/SD\_CLK, GPP\_G7/SD\_WP

### 3.3 V Operation

Input	VIH	Input High Voltage Threshold	0.7 x VCC	—	V		
	VIL	Input Low Voltage Threshold	—	0.3 x VCC	V		
	IIL	Input Leakage Current	-12	12	μA		
	CIN	Input Pin Capacitance	—	13	pF		
Output	VOH	Output High Voltage Threshold	VCC - 0.45	VCC	V	I <sub>OH</sub> =3 mA	
	VOL	Output Low Voltage Threshold	—	0.45	V	I <sub>OL</sub> = -3 mA	
	R <sub>pu</sub>	WPU Resistance	5K-70% 20K-25%	5K+70% 20K+25%	Ohm		
	R <sub>pd</sub>	WPD Resistance	5K-70% 20K-25%	5K+70% 20K+25%	Ohm		

### 1.8 V Operation

Input	VIH	Input High Voltage Threshold	0.7 x VCC	—	V		
	VIL	Input Low Voltage Threshold	—	0.3 x VCC	V		
	IIL	Input Leakage Current	-12	12	μA		
	CIN	Input Pin Capacitance	—	13	pF		
Output	V <sub>OH</sub>	Output High Voltage Threshold	VCC - 0.45	VCC	V	I <sub>OH</sub> =3 mA	
	V <sub>OL</sub>	Output Low Voltage Threshold	—	0.45	V	I <sub>OL</sub> = -3 mA	
	R <sub>pu</sub>	WPU Resistance	5K-70% 20K-25%	5K+70% 20K+25%	Ohm		
	R <sub>pd</sub>	WPD Resistance	5K-70% 20K-25%	5K+70% 20K+25%	Ohm		

Note: 1. For GPIO supported voltages, refer to [General Purpose Input and Output \(GPIO\)](#) on page 127.

**continued...**

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
<b>Note:</b> For GPIO pads (GPP) listed in the Associated Signals below, all functions that are multiplexed on GPIO pads will have the same DC characteristics as the GPIO pads. Refer to <a href="#">General Purpose Input and Output (GPIO)</a> on page 127 for the muxed functions on a specific GPIO pad. <b>Associated Signals:</b> GPP_K0, GPP_K1, GPP_K2, GPP_K3, GPP_K4, GPP_K5, GPP_K6, GPP_K7, GPP_K8, GPP_K9, GPP_K10, GPP_K11, GPP_K12/GSXSDOUT, GPP_K13/GSXSLDOUT, GPP_K14/GSXSDIN, GPP_K15/GSXSRRESET#, GPP_K16/GSXCLK, GPP_K17/ADR_COMPLETE, GPP_K18/NMI#, GPP_K19/SMI#, GPP_K20, GPP_K21, GPP_K22/IMGCLKOUT0, GPP_K23/IMGCLKOUT1, GPP_H0/SRCCLKREQ6#, GPP_H1/SRCCLKREQ7#, GPP_H2/SRCCLKREQ8#, GPP_H3/SRCCLKREQ9#, GPP_H4/SRCCLKREQ10#, GPP_H5/SRCCLKREQ11#, GPP_H6/SRCCLKREQ12#, GPP_H7/SRCCLKREQ13#, GPP_H8/SRCCLKREQ14#, GPP_H9/SRCCLKREQ15#, GPP_H10/SML2CLK, GPP_H11/SML2DATA, GPP_H12/SML2ALERT#, GPP_H13/SML3CLK, GPP_H14/SML3DATA, GPP_H15/SML3ALERT#, GPP_H16/SML4CLK, GPP_H17/SML4DATA, GPP_H18/SML4ALERT#, GPP_H19/ISH_I2C0_SDA, GPP_H20/ISH_I2C0_SCL, GPP_H21/ISH_I2C1_SDA, GPP_H22/ISH_I2C1_SCL, GPP_H23/TIME_SYNC0, GPP_I0/DDPB_HPD0/DISP_MISC0, GPP_I1/DDPC_HPD1/DISP_MISC1, GPP_I2/DDPD_HPD2/DISP_MISC2, GPP_I3/DDPF_HPD3/DISP_MISC3, GPP_I4/EDP_HPD/DISP_MISC4, GPP_I5/DDPB_CTRLCLK, GPP_I6/DDPB_CTRLDATA, GPP_I7/DDPC_CTRLCLK, GPP_I8/DDPC_CTRLDATA, GPP_I9/DDPD_CTRLCLK, GPP_I10/DDPD_CTRLDATA, GPP_I11/M2_SKT2_CFG0, GPP_I12/M2_SKT2_CFG1, GPP_I13/M2_SKT2_CFG2, GPP_I14/M2_SKT2_CFG3, SYS_PWROK, SYS_RESET#							
<b>3.3 V Operation</b>							
Input	V <sub>IH</sub>	Input High Voltage Threshold	0.75 x VCC	—	V		
	V <sub>IL</sub>	Input Low Voltage Threshold	—	0.25 x VCC	V		
	I <sub>IL</sub>	Input Leakage Current	-10	10	μA		
	C <sub>IN</sub>	Input Pin Capacitance	—	14	pF		
Output	V <sub>OH</sub>	Output High Voltage Threshold	VCC - 0.45	VCC	V	I <sub>OH</sub> =3 mA	
	V <sub>OL</sub>	Output Low Voltage Threshold	—	0.45	V	I <sub>OL</sub> =-3 mA	
	R <sub>pu</sub>	WPU Resistance	1K-50% 5K-70% 20K-35%	1K+100% 5K+70% 20K+35%	Ohm		
	R <sub>pd</sub>	WPD Resistance	5K-70% 20K-35%	5K+70% 20K+35%	Ohm		
<b>1.8 V Operation</b>							
Input	V <sub>IH</sub>	Input High Voltage Threshold	0.75 x VCC	—	V		
	V <sub>IL</sub>	Input Low Voltage Threshold	—	0.25 x VCC	V		
	I <sub>IL</sub>	Input Leakage Current	-10	10	μA		
	C <sub>IN</sub>	Input Pin Capacitance	—	14	pF		
Output	V <sub>OH</sub>	Output High Voltage Threshold	VCC - 0.45	VCC	V	I <sub>OH</sub> =3 mA	
	V <sub>OL</sub>	Output Low Voltage Threshold	—	0.45	V	I <sub>OL</sub> =-3 mA	
	R <sub>pu</sub>	WPU Resistance	1K-50% 5K-70% 20K-35%	1K+100% 5K+70% 20K+35%	Ohm		
	R <sub>pd</sub>	WPD Resistance	5K-70% 20K-35%	5K+70% 20K+35%	Ohm		
continued...							

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
<b>Note:</b> For GPIO pads (GPP) listed in the Associated Signals below, all functions that are multiplexed on GPIO pads will have the same DC characteristics as the GPIO pads. Refer to <a href="#">General Purpose Input and Output (GPIO)</a> on page 127 for the muxed functions on a specific GPIO pad. <b>Associated Signals:</b> GPP_E0/SATAXPCE0/SATAGP0, GPP_E1/SATAXPCE1/SATAGP1, GPP_E2/SATAXPCE2/SATAGP2, GPP_E3/CPU_GP0, GPP_E4/SATA_DEVSLP0, GPP_E5/SATA_DEVSLP1, GPP_E6/SATA_DEVSLP2, GPP_E7/CPU_GP1, GPP_E8/SATALED#, GPP_E9/USB2_OC0#, GPP_E10/USB2_OC1#, GPP_E11/USB2_OC2#, GPP_E12/USB2_OC3#, GPP_F0/SATAXPCE3/SATAGP3, GPP_F1/SATAXPCE4/SATAGP4, GPP_F2/SATAXPCE5/SATAGP5, GPP_F3/SATAXPCE6/SATAGP6, GPP_F4/SATAXPCE7/SATAGP7, GPP_F5/SATA_DEVSLP3, GPP_F6/SATA_DEVSLP4, GPP_F7/SATA_DEVSLP5, GPP_F8/SATA_DEVSLP6, GPP_F9/SATA_DEVSLP7, GPP_F10/SATA_SCLOCK, GPP_F11/SATA_SLOAD, GPP_F12/SATA_SDATAOUT1, GPP_F13/SATA_SDATAOUT0, GPP_F14/PS_ON#, GPP_F15/USB2_OC4#, GPP_F16/USB2_OC5#, GPP_F17/USB2_OC6#, GPP_F18/USB2_OC7#, GPP_F19/eDP_VDDEN, GPP_F20/eDP_BKLTEN, GPP_F21/eDP_BKLTCTL, GPP_F22/DDPF_CTRLCLK, GPP_F23/DDPF_CTRLDATA, SPI0_IO2, SPI0_IO3, GPP_B16/GSPI0_CLK							
<b>3.3 V Operation</b>							
Input	VIH	Input High Voltage Threshold	0.7 x VCC	—	V		
	VIL	Input Low Voltage Threshold	—	0.3 x VCC	V		
	IIL	Input Leakage Current	-12	12	μA		
	CIN	Input Pin Capacitance	—	13	pF		
Output	VOH	Output High Voltage Threshold	VCC - 0.45	VCC	V	I <sub>OH</sub> =3 mA	
	VOL	Output Low Voltage Threshold	—	0.45	V	I <sub>OL</sub> =-3 mA	
	R <sub>pu</sub>	WPU Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm		
	R <sub>pd</sub>	WPD Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm		
<b>1.8 V Operation</b>							
Input	VIH	Input High Voltage Threshold	0.7 x VCC	—	V		
	VIL	Input Low Voltage Threshold	—	0.3 x VCC	V		
	IIL	Input Leakage Current	-12	12	μA		
	CIN	Input Pin Capacitance	—	13	pF		
Output	V <sub>OH</sub>	Output High Voltage Threshold	VCC - 0.45	VCC	V	I <sub>OH</sub> =3 mA	
	V <sub>OL</sub>	Output Low Voltage Threshold	—	0.45	V	I <sub>OL</sub> =-3 mA	
	R <sub>pu</sub>	WPU Resistance	5K-70% 20K-25%	5K+70% 20K+35%	Ohm		
	R <sub>pd</sub>	WPD Resistance	5K-70% 20K-25%	5K+70% 20K+35%	ohm		
<b>Associated Signals:</b> PM_SYNC, Peci, CPUPWRGD, THRMTrip#, PLTRST_CPU#, PM_DOWN, TRIGGER_IN, TRIGGER_OUT, PCH_JTAG_TDO, PCH_JTAGX, PRDY#, PREQ#, CPU_TRST#, PCH_JTAG_TDI, PCH_JTAG_TMS, PCH_JTAG_TCK, ITP_PMODE, HDACPU_SDI, HDACPU_SCLK, HDACPU_SDO							
<b>1.05 V Operation</b>							
Input	VIH	Input High Voltage Threshold	JTAG: 0.8 x VCC	—	V		

*continued...*

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
			PECI: 0.7 x VCC iDISPLAY: 0.65 x VCC				
	VIL	Input Low Voltage Threshold	—	JTAG: 0.45 x VCC PECI: 0.3 x VCC iDISPLAY: 0.3 x VCC	V		
	IIL	Input Leakage Current	-10	10	μA		
	CIN	Input Pin Capacitance	—	2	pF		
Output	VOH	Output High Voltage Threshold	0.75 x VCC	VCC	V	I <sub>OH</sub> =6 mA	
	VOL	Output Low Voltage Threshold	—	0.25 x VCC	V	I <sub>OL(min)</sub> = -0.5 mA I <sub>OL(max)</sub> = -1 mA	
	R <sub>pu</sub>	WPU Resistance	1K-30% 20K-30%	1K+30% 20K+30%	Ohm		
	R <sub>pd</sub>	WPD Resistance	1K-30% 20K-30%	1K+30% 20K+30%	Ohm		
<b>Associated Signals:</b> GPP_J0/CNV_PA_BLANKING, GPP_J1/CPU_C10_GATE#, GPP_J2, GPP_J3, GPP_J4/CNV_BRI_DT/UART0B_RTS#, GPP_J5/CNV_BRI_RSP/UART0B_RXD, GPP_J6/CNV_RGI_DT/UART0B_TXD, GPP_J7/CNV_RGI_RSP/UART0B_CTS#, GPP_J8/CNV_MFUART2_RXD, GPP_J9/CNV_MFUART2_TXD, GPP_J10, GPP_J11/A4WP_PRESENT							
<b>3.3 V Operation</b>							
Input	VIH	Input High Voltage Threshold	0.65 x VCC	—	V		
	VIL	Input Low Voltage Threshold	—	0.35 x VCC	V		
	IIL	Input Leakage Current	-5	5	μA		
	CIN	Input Pin Capacitance	—	5	pF		
Output	VOH	Output High Voltage Threshold	VCC - 0.45	—	V	I <sub>OH</sub> =3 mA	
	VOL	Output Low Voltage Threshold	—	0.45	V	I <sub>OL</sub> = -3 mA	
	R <sub>pu</sub>	WPU Resistance	20K-40%	20K+40%	Ohm		
	R <sub>pd</sub>	WPD Resistance	20K-40%	20K+40%	Ohm		
<b>1.8 V Operation</b>							
Input	VIH	Input High Voltage Threshold	0.65 x VCC	—	V		
	VIL	Input Low Voltage Threshold	—	0.35 x VCC	V		
	IIL	Input Leakage Current	-5	5	μA		
	CIN	Input Pin Capacitance	—	5	pF		
Output	V <sub>OL</sub>	Output Low Voltage Threshold	VCC - 0.45	—	V	I <sub>OL</sub> = -3 mA	
continued...							

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
			—	0.45			
	R <sub>pu</sub>	WPU Resistance Threshold	20K-40%	20K+40%	Ohm		
	R <sub>pd</sub>	WPD Resistance	20K-40%	20K+40%	Ohm		
<b>Notes:</b> 1. The VOH specification does not apply to open-collector or open-drain drivers. Signals of this type must have an external pull-up resistor, and that is what determines the high-output voltage level. 2. Input characteristics apply when a signal is configured as Input or to signals that are only Inputs. Output characteristics apply when a signal is configured as an Output or to signals that are only Outputs.							

**Table 26. Single-Ended Signal DC Characteristics as Inputs or Outputs**

Type	Symbol	Parameter	Minimum	Maximum	Unit	Condition	Notes
<b>Associated Signals: INTRUDER#, RSMRST#, PCH_PWROK, DSW_PWROK, SRTCST#</b>							
Input	VIH	Input High Voltage Threshold	0.65 × VCCRTC	VCCRTC+0.5	V		1, 2
	VIL	Input Low Voltage Threshold	-0.5	0.3 × VCCRTC	V		2
<b>Associated Signals: RTCRST#</b>							
Input	VIH	Input High Voltage Threshold	0.75 × VCCRTC	VCCRTC+0.5	V		1, 2
	VIL	Input Low Voltage Threshold	-0.5	0.4 × VCCRTC	V		2
<b>Associated Signals: RTCX1#</b>							
Input	VIH	Input High Voltage Threshold	0.8	1.2	V		
	VIL	Input Low Voltage Threshold	-0.5	0.1	V		
<b>Notes:</b> 1. VCCRTC is the voltage applied to the VCCRTC well of the PCH. When the system is in G3 state, it is generally supplied by the coin cell battery. 2. These buffers have input hysteresis. VIH levels are for rising edge transitions and VIL levels are for falling edge transitions.							

**Table 27. Differential Signals Characteristics**

Symbol	Parameter	Minimum	Maximum	Unit	Conditions	Notes
<b>Associated Signals: PCIe</b>						
<b>Gen 1</b>						
VTX-DIFF P-P	Differential Peak to Peak Output Voltage	0.8	1.2	V		1
VTX-DIFF P-P - Low	Low power differential Peak to Peak Output Voltage	0.4	1.2	V		
VTX_CM-ACp	TX AC Common Mode Output Voltage (2.5 GT/s)	—	20	mV		
ZTX-DIFF-DC	DC Differential TX Impedance	80	120	Ohm		
VRX-DIFF p-p	Differential Input Peak to Peak Voltage	0.12	1.2	V		1
<b>continued...</b>						

Symbol	Parameter	Minimum	Maximum	Unit	Conditions	Notes
VRX_CM-ACp	AC peak Common Mode Input Voltage	—	150	mV		
<b>Gen 2</b>						
VTX-DIFF P-P	Differential Peak to Peak Output Voltage	0.8	1.2	V		
VTX-DIFF P-P - Low	Low power differential Peak to Peak Output Voltage	0.4	1.2	V		
VTX_CM-ACp-p	TX AC Common Mode Output Voltage (5 GT/s)	—	100	mV		
ZTX-DIFF-DC	DC Differential TX Impedance	80	120	Ohm		
VRX-DIFF p-p	Differential Input Peak to Peak Voltage	0.12	1.2	V		
VRX_CM-ACp	AC peak Common Mode Input Voltage	—	150	mV		
<b>Gen 3</b>						
VTX-DIFF P-P	Differential Peak to Peak Output Voltage	0.8	1.3	V		
VTX-DIFF P-P - Low	Low power differential Peak to Peak Output Voltage	0.4	1.2	V		
VTX_CM-ACp-p	TX AC Common Mode Output Voltage (8 GT/s)	—	100	mV		
ZTX-DIFF-DC	DC Differential TX Impedance	80	120	Ohm		
VRX-DIFF p-p	Differential Input Peak to Peak Voltage	Refer to Stressed Voltage Eye Parameters Table in PCIe GEN3 industry specifications.				
VRX_CM-ACp	AC peak Common Mode Input Voltage	—	150	mV		
<b>Associated Signals: SATA</b>						
VIMIN-Gen1i	Minimum Input Voltage -1.5 Gb/s internal SATA	325	—	mVdiff p-p		2
VIMAX-Gen1i	Maximum Input Voltage -1.5 Gb/s internal SATA	—	600	mVdiff p-p		2
VIMIN-Gen1m	Minimum Input Voltage -1.5 Gb/s eSATA	240	—	mVdiff p-p		2
VIMAX-Gen1m	Maximum Input Voltage -1.5 Gb/s eSATA	—	600	mVdiff p-p		2
VIMIN-Gen2i	Minimum Input Voltage -3.0 Gb/s internal SATA	275	—	mVdiff p-p		2
VIMAX-Gen2i	Maximum Input Voltage -3.0 Gb/s internal SATA	—	750	mVdiff p-p		2
VIMIN-Gen2m	Minimum Input Voltage -3.0 Gb/s eSATA	240	—	mVdiff p-p		2
VIMAX-Gen2m	Maximum Input Voltage -3.0 Gb/s eSATA	—	750	mVdiff p-p		2
VIMIN-Gen3i	Minimum Input Voltage -6.0 Gb/s internal SATA	240	—	mVdiff p-p		2
<b>continued...</b>						

Symbol	Parameter	Minimum	Maximum	Unit	Conditions	Notes
VIMAX-Gen3i	Maximum Input Voltage -6.0 Gb/s internal SATA	—	1000	mVdiff p-p		2
VOMIN-Gen1i,m	Minimum Output Voltage-1.5 Gb/s internal and eSATA	400	—	mVdiff p-p		3
VOMAX-Gen1i,m	Maximum Output Voltage-1.5 Gb/s internal and eSATA	—	600	mVdiff p-p		3
VOMIN-Gen2i,m	Minimum Output Voltage-3.0 Gb/s internal and eSATA	400	—	mVdiff p-p		3
VOMAX-Gen2i,m	Maximum Output Voltage-3.0 Gb/s internal and eSATA	—	700	mVdiff p-p		3
VOMIN-Gen3i	Minimum Output Voltage-6.0 Gb/s internal SATA	200	—	mVdiff p-p		3
VOMAX-Gen3i	Maximum Output Voltage-6.0 Gb/s internal SATA	—	900	mVdiff p-p		3
<b>Associated Signals: USB 2.0</b>						
VDI	Differential Input Sensitivity	0.2	—	V		4, 6
VCM	Differential Common Mode Range	0.8	2.5	V		5, 6
VSE	Single-Ended Receiver Threshold	0.8	2	V		6
VCRS	Output Signal Crossover Voltage	1.3	2	V		6
VOL	Output Low Voltage	—	0.4	V	I <sub>OL</sub> =5 mA	6
VOH	Output High Voltage	3.3 V – 0.5	—	V	I <sub>OH</sub> = -2 mA	6
VHSSQ	HS Squelch Detection Threshold	100	150	mV		7
VHSDSC	HS Disconnect Detection Threshold	525	625	mV		7
VHSCM	HS Data Signaling Common Mode Voltage Range	-50	500	mV		7
VHSOI	HS Idle Level	-10	10	mV		7
VHSOH	HS Data Signaling High	360	440	mV		7
VHSOL	HS Data Signaling Low	-10	10	mV		7
VCHIRPJ	Chirp J Level	700	1100	mV		7
VCHIRPK	Chirp K Level	-900	-500	mV		7
<b>Associated Signals: USB 3.2</b>						
VTX-DIFF-PP	Differential Peak to Peak Output Voltage	0.8	1.2	V		
VTX-DIFF P-P - Low	Low power differential Peak to Peak Output Voltage	0.4	1.2	V		8
VTX_CM-Acp-p	TX AC Common Mode Output Voltage (5 GT/s)	—	100	mV		
ZTX-DIFF-DC	DC Differential TX Impedance	72	120	Ohm		
						<b>continued...</b>

Symbol	Parameter	Minimum	Maximum	Unit	Conditions	Notes
VRX-DIFF p-p	Differential Input Peak to Peak Voltage	0.1	1.2	V		
VRX_CM-ACp	AC peak Common Mode Input Voltage	—	150	mV		
<b>Associated Signals: CLKOUT_CPUBCLK_P/N, CLKOUT_CPUPCIBCLK_P/N, CLKOUT_ITPXDP_P/N, CLKOUT_PCIE_P/N[15:0]</b>						
Output	VSwing	Differential Output Swing	300	—	mV	12
	VCross	Crossing Point Voltage	250	550	mV	11,13,14
	VCross_Delta	Variation of VCROSS	—	140	mV	11,13,14
	VMax	Maximum Output Voltage	—	1.15	V	11
	VMin	Minimum Output Voltage	-0.3	—	V	11
<b>Notes:</b> 1. PCI Express* mVdiff p-p = $2 *  PCIE[x]_{TXP} - PCIE[x]_{TXN} $ ; PCI Express mVdiff p-p = $2 *  PCIE[x]_{RXP} - PCIE[x]_{RXN} $ 2. SATA Vdiff, RX ( $V_{IMAX}/V_{IMIN}$ ) is measured at the SATA connector on the receiver side (generally, the motherboard connector), where SATA mVdiff p-p = $2 *  SATA[x]_{RXP} - SATA[x]_{RXN} $ . 3. SATA Vdiff, TX ( $V_{OMIN}/V_{OMAX}$ ) is measured at the SATA connector on the transmit side (generally, the motherboard connector), where SATA mVdiff p-p = $2 *  SATA[x]_{TXP} - SATA[x]_{TXN} $ 4. $V_{DI} =  USBPx[P] - USBPx[N] $ 5. Includes VDI range 6. Applies to Low-Speed/Full-Speed USB 2.0 7. Applies to High-Speed USB 2.0. 8. USB 3.2 mVdiff p-p = $2 *  USB3Rp[x] - USB3Rn[x] $ ; USB 3.2 mVdiff p-p = $2 *  USB3Tp[x] - USB3Tn[x] $ 9. For PCIe, Gen 1, Gen 2 and Gen 3 correspond to the PCIe base specification revision 1, 2 and 3. 10. PCIe specifications are also applicable to the LAN port 11. Measurement taken from single-ended waveform on a component test board 12. Measurement taken from differential waveform on a component test board 13. VCross is defined as the voltage where Clock = Clock# 14. Only applies to the differential rising edge (that is, Clock rising and Clock# falling) 15. The total variation of all VCross measurements in any particular system. Note that this is a subset of VCross MIN/MAX (VCross absolute) allowed. The intent is to limit VCross induced modulation by setting VCross_Delta to be smaller than VCross absolute.						

**Table 28. Power Rail Operational Voltage Range**

Symbol	Parameter	Minimum	Nominal	Maximum	Unit	Notes
VCCPRIM_1p05	Core Logic, SRAM, I/O Blocks, USB AFE, Processor Sideband, JTAG, Thermal Sensor Primary WellSP, Analog Supply for Audio PLL Primary Well, Mod-PHY Supply Primary Well	0.9975	1.05	1.1025	V	1
VCCPRIM_3p3	Primary Well for HVC MOS and display, USB 2.0, Integrated 1.8 V VRM input supply	3.135	3.3	3.465	V	1
VCCPRIM_1P8	Primary Well for 1.8 V GPIO and biasing	1.71	1.8	1.89	V	1
VCCPRIM_MPHY_1P05	Primary supply for PCIe/USB PLL dividers	0.9975	1.05	1.1025	V	1
VCCAMPHYPLL_1p05	Analog Supply for USB 3.2, PCIe Gen2, SATA and PCIe Gen 3 PLL Primary Well	0.9975	1.05	1.1025	V	1
continued...						

Symbol	Parameter	Minimum	Nominal	Maximum	Unit	Notes
VCCA_BCLK_1P05	Analog supply for BCLK buffer	0.9975	1.05	1.1025	V	1
VCCA_SRC_1P05	Analog supply for PCIe clock buffer	0.9975	1.05	1.1025	V	1
VCCA_XTAL_1P05	Analog supply for XTAL circuitries	0.9975	1.05	1.1025	V	1
VCCAPLL_1P05	Analog supply for BCLK PLLs	0.9975	1.05	1.1025	V	1
VCCDPHY_1P24	1.24 V for CNVi logic (Internal generated)	1.178	1.24	1.302	V	1
VCCDSW_1P05	Deep Sx Well (Internal generated)	0.9975	1.05	1.1025	V	1
VCCDUSB_1P05	Supply for USB AFE	0.9975	1.05	1.1025	V	1
VCCPHVLD0_1P8	Integrated 1.8 V VRM output	1.71	1.8	1.89	V	1
VCCPGPPA	Group A Primary Well GPIOs	3.135	3.3	3.465	V	1
		1.71	1.8	1.89	V	1
VCCPGPPBC	Group B and C Primary Well GPIOs	3.135	3.3	3.465	V	1
		1.71	1.8	1.89	V	1
VCCPGPPD	Group D Primary Well GPIOs	3.135	3.3	3.465	V	1
		1.71	1.8	1.89	V	1
VCCPGPPEF	Group E and F Primary Well GPIOs	3.135	3.3	3.465	V	1
		1.71	1.8	1.89	V	1
VCCPGPPG_3P3	Group G Primary Well GPIOs	3.135	3.3	3.465	V	1
VCCPGPPHK	Group H and K Primary Well GPIOs	3.135	3.3	3.465	V	1
		1.71	1.8	1.89	V	1
VCCSPI	SPI Primary Well	3.135	3.3	3.465	V	1
		1.71	1.8	1.89	V	1
VCCHDA	Intel® HD Audio Supply Primary Well	3.135	3.3	3.465	V	1
		1.71	1.8	1.89	V	1
		1.425	1.5	1.575	V	1
VCCDSW_3p3	Deep Sx Well for GPD and USB 2.0	3.135	3.3	3.465	V	1
VCCRTC	RTC Well Supply	2	3.3	3.465	V	1,2
<b>Notes:</b> 1. The I/O buffer supply voltage is measured at the PCH package pins. The tolerances shown in this table are inclusive of all noise from DC up to 1 MHz. In testing, the voltage rails should be measured with a bandwidth limited oscilloscope that has a roll off of 3 db/decade above 20 MHz. 2. Maximum Crystal ESR is 50 kOhm.						

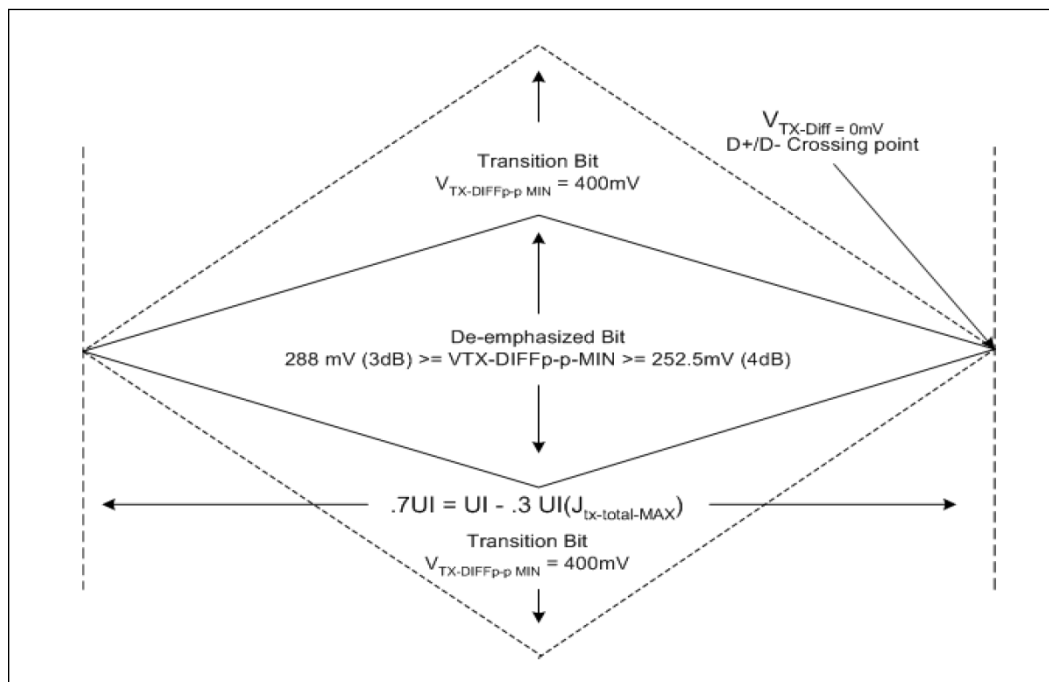
## 9.4 AC Characteristics

**Table 29. PCI Express\* Interface Timings**

Symbol	Parameter	Minimum	Maximum	Unit	Figures	Notes
<b>Transmitter and Receiver Timings</b>						
UI	Unit Interval – PCI Express*	399.88	400.12	ps		5
<b>continued...</b>						

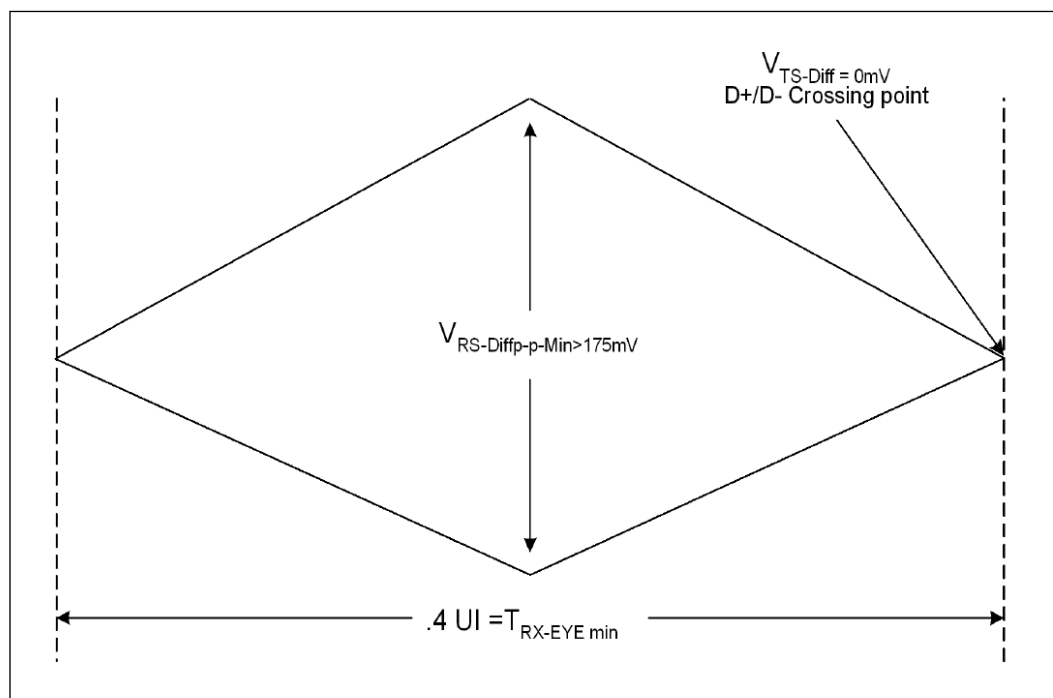
Symbol	Parameter	Minimum	Maximum	Unit	Figures	Notes
<b>Transmitter and Receiver Timings</b>						
(Gen1)						
UI (Gen 2)	Unit Interval – PCI Express	199.9	200.1	ps		5
UI (Gen 3)	Unit Interval – PCI Express	124.96	125.03	ps		
T <sub>TX-EYE</sub> (Gen 1/ Gen 2)	Minimum Transmission Eye Width	0.75	—	UI	Figure 4 on page 70	1,2
T <sub>TX-EYE-MEDIAN- to-MAX-JITTER</sub> (Gen 1)	Maximum time between the jitter median and maximum deviation from the median	0.125	—	UI		1,2
T <sub>TX-EYE-MEDIAN- to-MAX-JITTER</sub> (Gen 2)	Maximum time between the jitter median and maximum deviation from the median	0.15	—	UI		
T <sub>TX-EYE-MEDIAN- to-MAX-JITTER</sub> (Gen 3)	Maximum time between the jitter median and maximum deviation from the median	0.15	—	UI		
T <sub>RX-EYE</sub> (Gen 1)	Minimum Receiver Eye Width	0.40	—	UI	Figure 5 on page 71	3,4
T <sub>RX-EYE</sub> (Gen 2)	Minimum Receiver Eye Width	0.60	—	UI	Figure 5 on page 71	3,4
T <sub>Min-Pulse</sub> (Gen 2)	Instantaneous Pulse Width	0.9	—	UI		
<p><b>Note:</b> Refer to <a href="http://www.pcisig.com">www.pcisig.com</a> for the updated specifications.</p> <ol style="list-style-type: none"> <li>Specified at the measurement point into a timing and voltage compliance test load and measured over any 250 consecutive TX UIs. (also refer to the Transmitter compliance eye diagram)</li> <li>A TTX-EYE = 0.70 UI provides for a total sum of deterministic and random jitter budget of TTXJITTER-MAX = 0.30 UI for the Transmitter collected over any 250 consecutive TX UIs. The TTXEYE-MEDIAN-to-MAX-JITTER specification ensures a jitter distribution in which the median and the maximum deviation from the median is less than half of the total TX jitter budget collected over any 250 consecutive TX UIs. It should be noted that the median is not the same as the mean. The jitter median describes the point in time where the number of jitter points on either side is approximately equal as opposed to the averaged time value.</li> <li>Specified at the measurement point and measured over any 250 consecutive UIs. The test load documented in the PCI Express* specification 2.0 should be used as the RX device when taking measurements (also refer to the Receiver compliance eye diagram). If the clocks to the RX and TX are not derived from the same reference clock, the TX UI recovered from 3500 consecutive UI must be used as a reference for the eye diagram.</li> <li>A TRX-EYE = 0.40 UI provides for a total sum of 0.60 UI deterministic and random jitter budget for the Transmitter and interconnect collected any 250 consecutive UIs. The TRX-EYE-MEDIAN-to-MAX-JITTER specification ensures a jitter distribution in which the median and the maximum deviation from the median is less than half of the total 0.6 UI jitter budget collected over any 250 consecutive TX UIs. It should be noted that the median is not the same as the mean. The jitter median describes the point in time where the number of jitter points on either side is approximately equal as opposed to the averaged time value. If the clocks to the RX and TX are not derived from the same reference clock, the TX UI recovered from 3500 consecutive UI must be used as the reference for the eye diagram.</li> <li>Nominal Unit Interval is 400 ps for 2.5 GT/s and 200 ps for 5 GT/s.</li> </ol>						

Figure 4. PCI Express\* Transmitter Eye



**NOTE**

Gen 1 example is shown for the illustration. Refer to [www.pcisig.com](http://www.pcisig.com) for the updated specifications.

**Figure 5. PCI Express\* Receiver Eye****NOTE**

Gen 1 example is shown for the illustration. Refer to [www.pcisig.com](http://www.pcisig.com) for the updated specifications.

**Table 30. DDC Characteristics**

Signal Group: eDP_VDDEN, eDP_BKLTEN, eDP_BKLTCTL, DDP[F,D:B]_CTRLCLK, DDP[F,D:B]_CTRLDATA							
Symbol	Parameter	Standard Mode	Fast Mode		1 MHz		Units
		Maximum	Minimum	Maximum	Minimum	Maximum	
$F_{scl}$	Operating Frequency	100	0	400	0	1000	kHz
$T_r$	Rise Time <sup>1</sup>	1000	$20 + 0.1Cb^2$	300	—	120	ns
$T_f$	Fall Time <sup>1</sup>	300	$20 + 0.1Cb^2$	300	—	120	ns
Notes: 1. Measurement Point for Rise and Fall time: $V_{IL(max)} - V_{IH(min)}$ 2. $C_b$ = total capacitance of one bus line in pF. If mixed with High-speed mode devices, faster fall times according to High-Speed mode $T_r/T_f$ are allowed.							

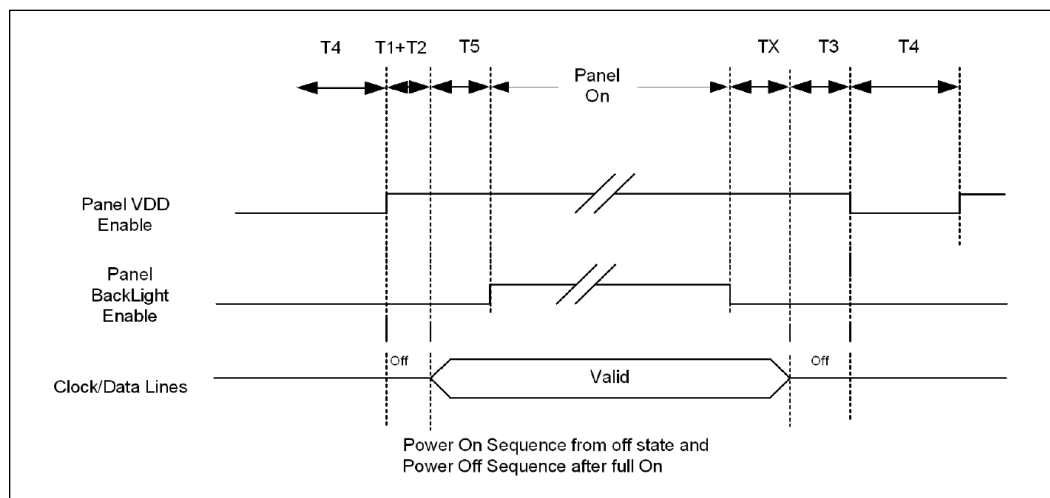
**9.4.1 Panel Power Sequencing and Backlight Control**

The PCH integrates panel power sequencing and backlight control signals for eDP\* interfaces on the processor.

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

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

**Figure 6. Panel Power Sequencing**



**NOTE**

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

**Table 31. DisplayPort\* Hot-Plug Detect Interface**

Signal Group: DDPB_HPD0, DDPC_HPD1, DDPD_HPD2, DDPF_HPD3, eDP_HPD						
Symbol	Parameter	Minimum	Maximum	Unit	Figures	Notes
Tir	Input Time Rise	50	500	ps		
Tif	Input Time Fall	50	500	ps		
Tidr	Input Delay Rise	0.3	2.5	ns		
Tidf	Input Delay Fall	0.3	2.5	ns		

**Table 32. Clock Timings**

Symbol	Parameter	Minimum	Maximum	Unit	Notes	Figure
LPC Clock (CLKOUT_LPC[1:0])						
	Period	41.16	42.18	ns		Figure 7 on page 76
continued...						

Symbol	Parameter	Minimum	Maximum	Unit	Notes	Figure
	High Time	16.67	25.00	ns		<a href="#">Figure 7</a> on page 76
	Low Time	16.67	25.00	ns		<a href="#">Figure 7</a> on page 76
	Duty Cycle	40	60	%		
	Jitter	—	500	ps	8, 9	
	Flight Time (PCH to Device)	—	3	ns		
<b>CLKOUT_PCIE_P/N[15:0], CLKOUT_ITPXD_P[N]</b>						
Clock Period	Period SSC On	9.849	10.201	ns		<a href="#">Figure 8</a> on page 77
Clock Period	Period SSC Off	9.849	10.151	ns		<a href="#">Figure 8</a> on page 77
Positive Duty Cycle (Differential), Negative Duty Cycle (Differential)	Duty Cycle	40	60	%		<a href="#">Figure 8</a> on page 77
	Differential Output Swing	300	—	mV		<a href="#">Figure 8</a> on page 77
Rising Edge Rate	Rising Edge Rate	1.5	4	V/ns		<a href="#">Figure 8</a> on page 77
Falling Edge Rate	Falling Edge Rate	1.5	4	V/ns		<a href="#">Figure 8</a> on page 77
	Jitter	—	150	ps	8,9,10	
SSC	Spread Spectrum	0	0.5	%	11	
<b>CLKOUT_CPUNSSC_P/N</b>						
Rising Edge Rate	Rising Edge Rate	1.3	4	V/ns		<a href="#">Figure 8</a> on page 77
Falling Edge Rate	Falling Edge Rate	1.3	4	V/ns		<a href="#">Figure 8</a> on page 77
	Slew Rate Matching	—	20	%		<a href="#">Figure 8</a> on page 77
Positive Duty Cycle (Differential), Negative Duty Cycle (Differential)	Duty Cycle	45	55	%		<a href="#">Figure 8</a> on page 77
	Jitter	—	6	ps RMS		<a href="#">Figure 8</a> on page 77
<b>CLKOUT_CPUPCIBLK_P/N, CLKOUT_CPUBCLK_P/N</b>						
Rising Edge Rate	Rising Edge Rate	1.3	4	V/ns		<a href="#">Figure 8</a> on page 77
Falling Edge Rate	Falling Edge Rate	1.3	4	V/ns		<a href="#">Figure 8</a> on page 77

*continued...*

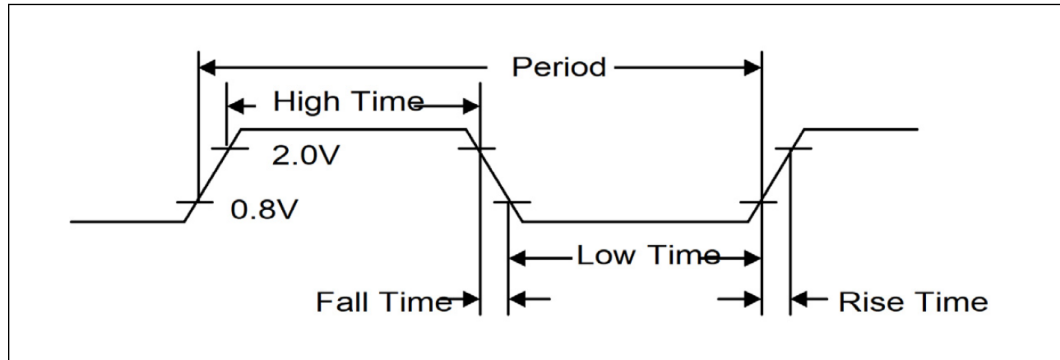
Symbol	Parameter	Minimum	Maximum	Unit	Notes	Figure
	Slew Rate Matching	—	20	%		Figure 8 on page 77
Positive Duty Cycle (Differential), Negative Duty Cycle (Differential)	Duty Cycle	40	60	%		Figure 8 on page 77
	Jitter	—	150	ps		Figure 8 on page 77
<b>SMBus/SMLink Clock (SMBCLK, SML[1:0]CLK)</b>						
fsmb	Operating Frequency	10	100	kHz		
t18	High Time	4.0	50	μs	2	Figure 9 on page 78
t19	Low Time	4.7	—	μs		Figure 9 on page 78
t20	Rise Time	—	1000	ns		Figure 9 on page 78
t21	Fall Time	—	300	ns		Figure 9 on page 78
<b>SMLink[1,0] (SML[1:0]CLK) (Fast Mode: Refer to note 15)</b>						
fsmb	Operating Frequency	0	400	kHz		
t18_SMLFM	High Time	0.6	50	μs	2	Figure 9 on page 78
t19_SMLFM	Low Time	1.3	—	μs		Figure 9 on page 78
t20_SMLFM	Rise Time	—	300	ns		Figure 9 on page 78
t21_SMLFM	Fall Time	—	300	ns		Figure 9 on page 78
<b>SMLink[1,0] (SML[1,0]CLK) (Fast Mode Plus: Refer to note 15)</b>						
fsmb	Operating Frequency	0	1000	kHz		
t18_SMLFMP	High Time	0.26	—	μs	2	Figure 9 on page 78
t19_SMLFMP	Low Time	0.5	—	μs		Figure 9 on page 78
t20_SMLFMP	Rise Time	—	120	ns		Figure 9 on page 78
t21_SMLFMP	Fall Time	—	120	ns		Figure 9 on page 78
<b>HDA_BLK (Intel® High Definition Audio)</b>						
f <sub>HDA</sub>	Operating Frequency	24.0		MHz		
	Frequency Tolerance	—	100	ppm		
	Input Jitter (refer to Clock Chip Specification)	—	300	ppm		
<b>continued...</b>						

Symbol	Parameter	Minimum	Maximum	Unit	Notes	Figure
	High Time (Measured at 0.75 VCC)	18.75	22.91	ns		Figure 7 on page 76
	Low Time (Measured at 0.35 VCC)	18.75	22.91	ns		Figure 7 on page 76
Suspend Clock (SUSCLK)						
fsusclk	Operating Frequency	32		kHz	4	
t39	High Time	9.5	—	μs	4	
t39a	Low Time	9.5	—	μs	4	
XTAL_IN/XTAL_OUT						
ppm12	Crystal Tolerance cut accuracy maximum	35 ppm(@ 25 °C ±3 °C)				
ppm12	Temp Stability Maximum	30 ppm(10 – 70 °C)				
ppm12	Aging Maximum	5 ppm				
Notes: 1. N/A						
2. The maximum high time (t18 Max.) provide a simple ensured method for devices to detect bus idle conditions.						
3. BCLK Rise and Fall times are measured from 10% VDD and 90% VDD.						
4. SUSCLK duty cycle can range from 30% minimum to 70% maximum.						
5. Edge rates in a system as measured from 0.8 – 2.0 V.						
6. The active frequency can be 5 MHz, 50 MHz, or 62.5 MHz depending on the interface speed. Dynamic changes of the normal operating frequency are not allowed.						
7. Testing condition: 1 kohm Pull-up to VCC, 1 kohm Pull-down and 10 pF Pull-down and 1/2 inch trace.						
8. Jitter is specified as cycle-to-cycle as measured between two rising edges of the clock being characterized. Period minimum and maximum includes cycle-to-cycle jitter and is also measured between two rising edges of the clock being characterized.						
9. On all jitter measurements care should be taken to set the zero crossing voltage (for rising edge) of the clock to be the point where the edge rate is the fastest. Using a Math function = Average(Derivative(Ch1)) and set the averages to 64, place the cursors where the slope is the highest on the rising edge—usually this lower half of the rising edge. The reason this is defined is for users trying to measure in a system it is impossible to get the probe exactly at the end of the Transmission line. This results in a reflection induced ledge in the middle of the rising edge and will significantly increase measured jitter.						
10. Phase jitter requirement: The designated outputs will meet the reference clock jitter requirements from the PCI Express Base Specification. The test is to be performed on a component test board under quiet conditions with all clock outputs on. Jitter analysis is performed using a standardized tool provided by the PCI SIG. Measurement methodology is defined in the Intel document “PCI Express Reference Clock Jitter Measurements”. This is not for CLKOUT_ITPXD_P/N.						
11. Spread Spectrum (SSC) is referenced to rising edge of the clock.						
12. Total of crystal cut accuracy, frequency variations due to temperature, parasitics, load capacitance variations and aging is recommended to be less than 90 ppm.						
13. Spread Spectrum (SSC) is referenced to rising edge of the clock.						
14. Spread Spectrum (SSC) of 0.25% on CLKOUT_PCIE[15:0] and CLKOUT_PEG_[B:A] is used for WiMAX friendly clocking purposes.						
15. When SMLink[1,0] is configured to run in Fast Mode Plus (FMP) using a soft strap, the supported operating range is 0 Hz ~ 1 MHz, but the typical operating frequency is in the range of 900 kHz – 1000 kHz. This is the default mode for this interface.						

**NOTE**

Refer to PCI Local Bus Specification for measurement details.

**Figure 7. Clock Timing**



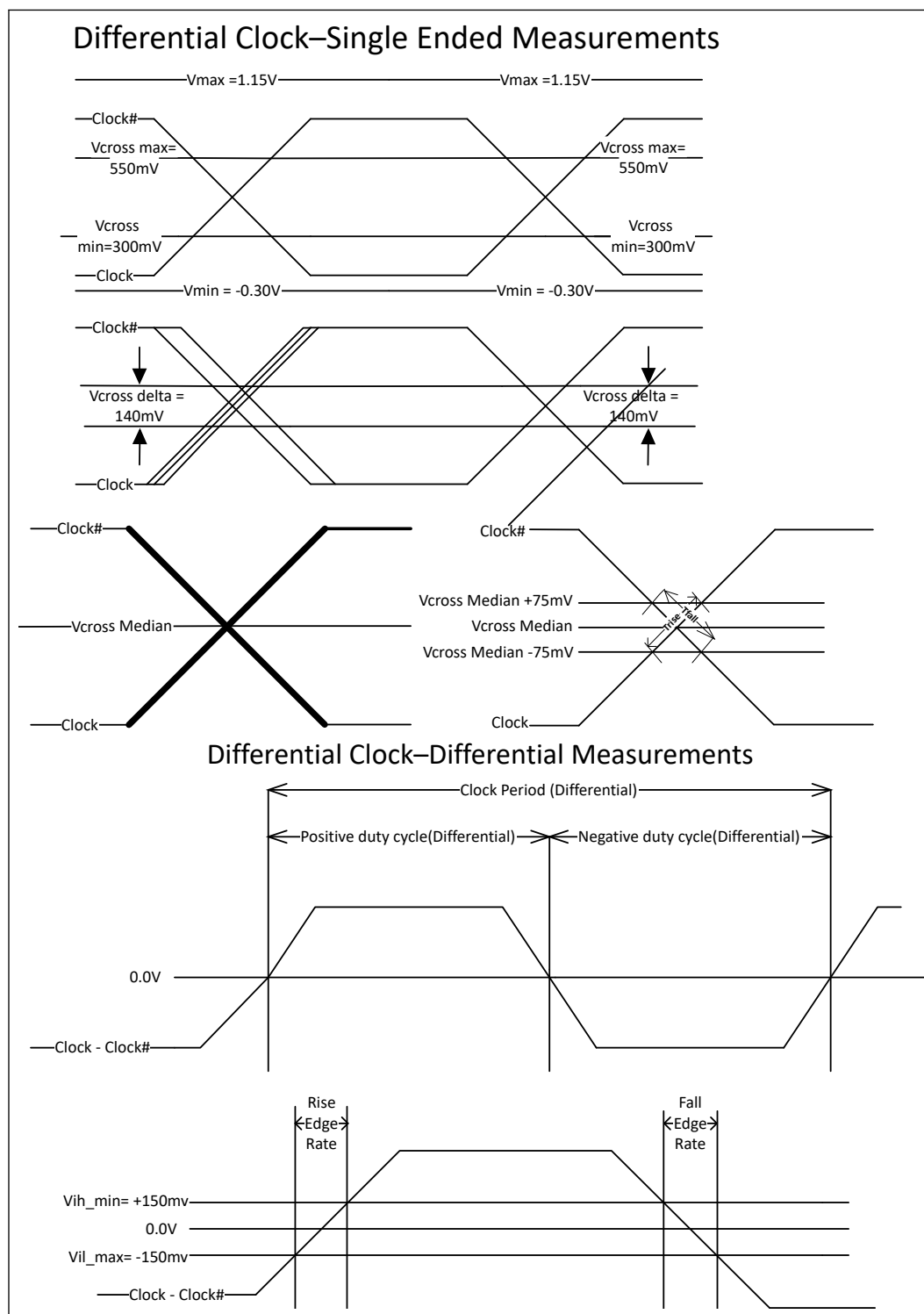
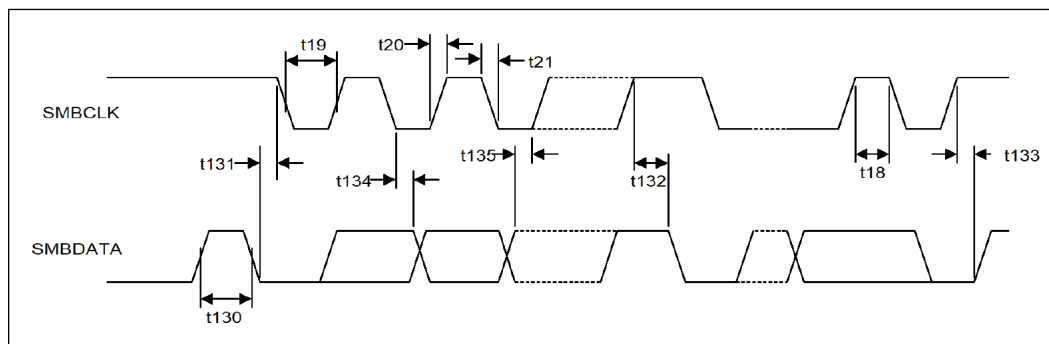
**Figure 8. Measurement Points for Differential Waveforms**


Figure 9. SMBus/SMLink Transaction



**NOTE**

txx also refers to txx\_SM and txx\_SMLFMP, txxx also refers to txxxSMLFM, SMBCLK also refers to SML[1:0]CLK, and SMBDATA also refers to SML[1:0]DATA.

Table 33. USB 2.0 Timing

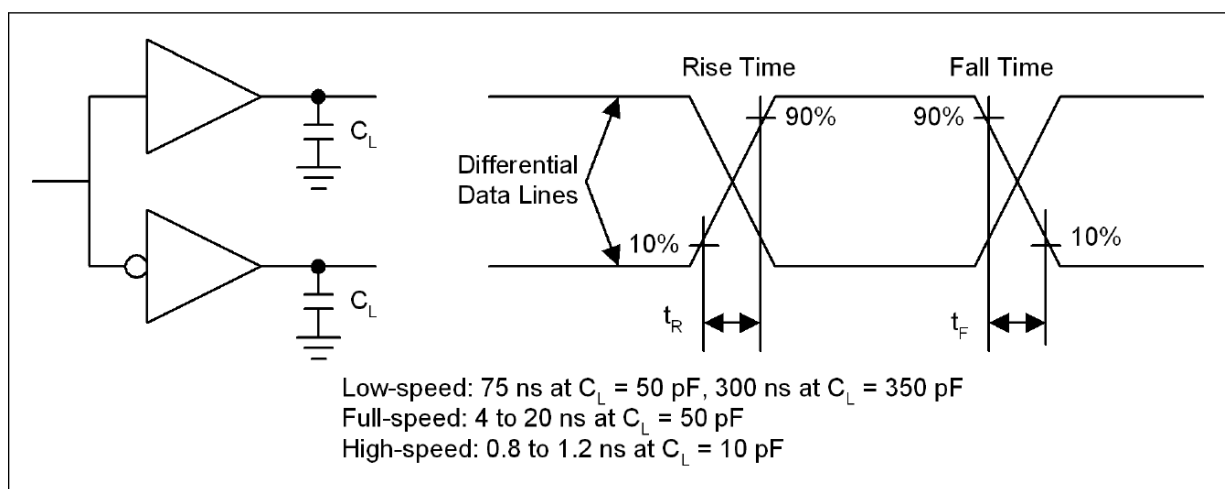
Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
<b>Full-speed Source (Refer to Note 7)</b>						
t100	USBPx+, USBPx- Driver Rise Time	4	20	ns	1,6 CL = 50 pF	Figure 10 on page 79
t101	USBPx+, USBPx- Driver Fall Time	4	20	ns	1,6 CL = 50 pF	Figure 10 on page 79
t102	Source Differential Driver Jitter - To Next Transition - For Paired Transitions	-3.5 -4	3.5 4	ns ns	2, 3	Figure 11 on page 80
t103	Source SE0 interval of EOP	160	175	ns	4	Figure 12 on page 80
t104	Source Jitter for Differential Transition to SE0 Transition	-2	5	ns	5	
t105	Receiver Data Jitter Tolerance - To Next Transition - For Paired Transitions	-18.5 -9	18.5 9	ns ns	3	Figure 11 on page 80
t106	EOP Width: Receiver must accept EOP	82	—	ns	4	Figure 12 on page 80
t107	Width of SE0 interval during differential transition	—	14	ns		
<b>Low-Speed Source (Refer to Note 8)</b>						
t108	USBPx+, USBPx - Driver Rise Time	75	300	ns	1,6 CL = 50 pF CL = 350 pF	Figure 10 on page 79
t109	USBPx+, USBPx - Driver Fall Time	75	300	ns	1,6 CL = 200 pF CL = 600 pF	Figure 10 on page 79
t110	Source Differential Driver Jitter	-25 -14	25 14	ns ns	2,3	Figure 11 on page 80

continued...

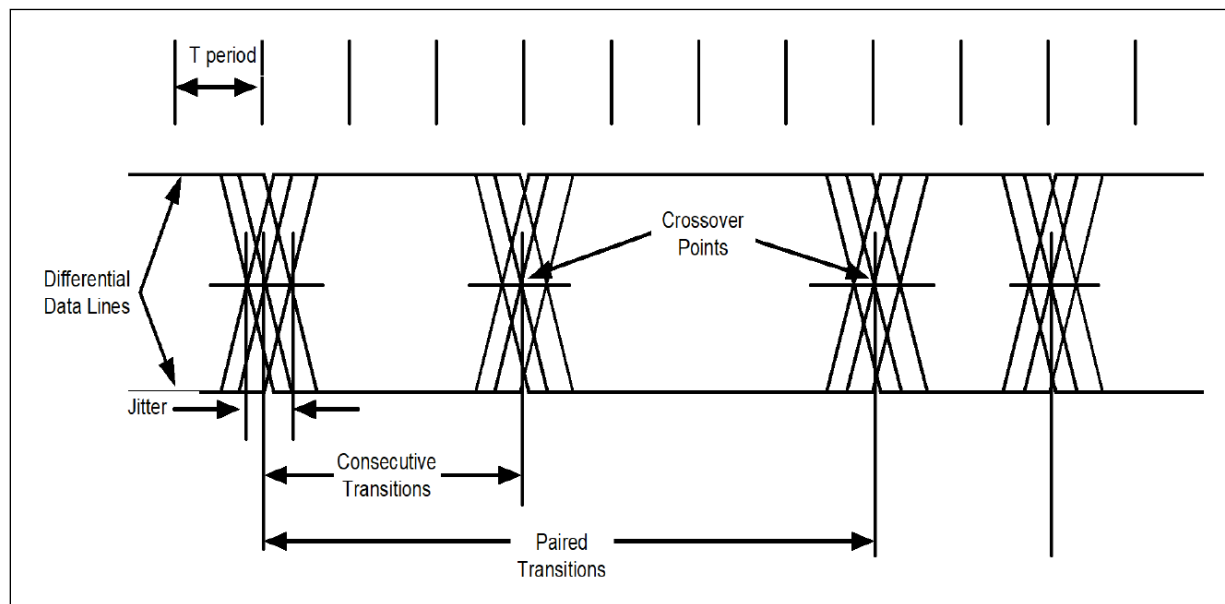
Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
	- To Next Transition - For Paired Transitions					
t111	Source SE0 interval of EOP	1.25	1.50	μs	4	Figure 12 on page 80
t112	Source Jitter for Differential Transition to SE0 Transition	-40	100	ns	5	
t113	Receiver Data Jitter Tolerance - To Next Transition - For Paired Transitions	-152 -200	152 200	ns ns	3	Figure 11 on page 80
t114	EOP Width: Receiver must accept EOP	670	—	ns	4	Figure 12 on page 80
t115	Width of SE0 interval during differential transition	—	210	ns		
<b>Notes:</b> 1. Driver output resistance under steady state drive is specified at 28 Ω at minimum and 43 Ω at maximum. 2. Timing difference between the differential data signals. 3. Measured at crossover point of differential data signals. 4. Measured at 50% swing point of data signals. 5. Measured from last crossover point to 50% swing point of data line at leading edge of EOP. 6. Measured from 10% to 90% of the data signal. 7. Full-speed Data Rate has minimum of 11.97 Mb/s and maximum of 12.03 Mb/s. 8. Low-speed Data Rate has a minimum of 1.48 Mb/s and a maximum of 1.52 Mb/s.						

**Table 34. USB 3.2 Interface Transmit and Receiver Timings**

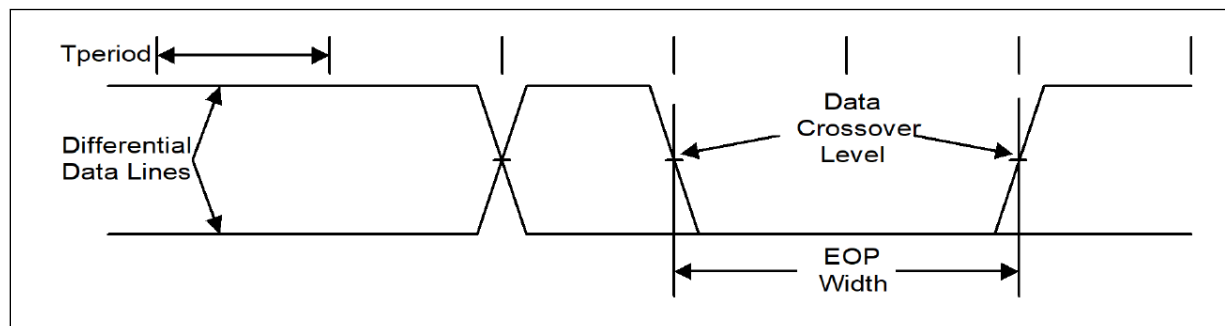
Symbol	Parameter	USB 3.2 Gen 1x1 (5 Gb/s)		USB 3.2 Gen 2x1 (10 Gb/s)		Units
		Minimum	Maximum	Minimum	Maximum	
UI	Unit Interval	199.94	200.06	99.97	100.03	ps
T <sub>TX-EYE</sub>	Minimum Transmission Eye Width	0.625	—	0.646	—	UI
PU3	Polling Period U3 State	—	100	—	100	ms
PRX-Detect	Polling Period Rx Detect	—	100	—	100	ms

**Figure 10. USB Rise and Fall Times**

**Figure 11. USB Jitter**



**Figure 12. USB EOP Width**



**Table 35. SATA Interface Timings**

Parameter	Minimum	Maximum	Units	Notes
SATA Operating Data Period (6 GB/s)	166.6083	166.6667	ps	
Rise Time	0.2	0.48	UI	1
Fall Time	0.2	0.48	UI	2
TX differential skew	—	20	ps	
COMINIT/COMRESET Transmit Gap Length	310.4	329.6	ns	3
COMWAKE Transmit Gap Length	103.5	109.9	ns	3
OOB Operating Data period	646.67	686.67	ns	4

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

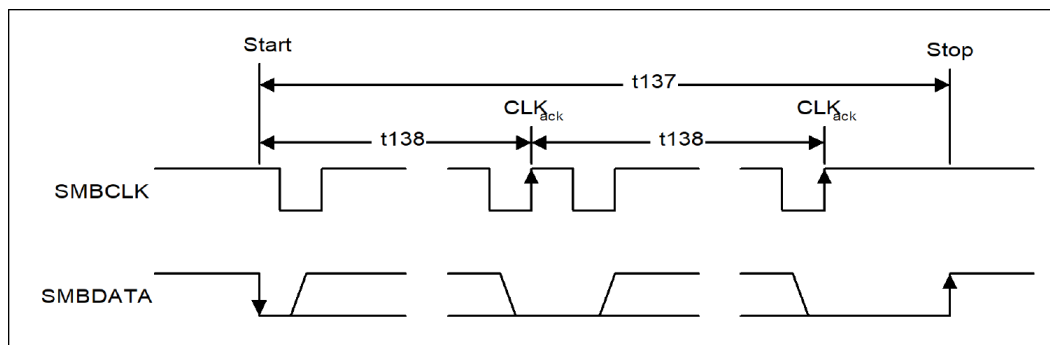
**Table 36. SMBus and SMLink Timing**

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t130	Bus Free Time Between Stop and Start Condition	4.7	—	μs		Figure 9 on page 78
t130SMLFM	Bus Free Time Between Stop and Start Condition	1.3	—	μs	5	Figure 9 on page 78
t130SMLFMP	Bus Free Time Between Stop and Start Condition	0.5	—	μs	5	Figure 9 on page 78
t131	Hold Time after (repeated) Start Condition. After this period, the first clock is generated.	4.0	—	μs		Figure 9 on page 78
t131SMLFM	Hold Time after (repeated) Start Condition. After this period, the first clock is generated.	0.6	—	μs	5	Figure 9 on page 78
t131SMLFMP	Hold Time after (repeated) Start Condition. After this period, the first clock is generated.	0.26	—	μs	5	Figure 9 on page 78
t132	Repeated Start Condition Setup Time	4.7	—	μs		Figure 9 on page 78
t132SMLFM	Repeated Start Condition Setup Time	0.6	—	μs	5	Figure 9 on page 78
t132SMLFMP	Repeated Start Condition Setup Time	0.26	—	μs	5	Figure 9 on page 78
t133	Stop Condition Setup Time	4.0	—	μs		Figure 9 on page 78
t133SMLFM	Stop Condition Setup Time	0.6	—	μs	5	Figure 9 on page 78
t133SMLFMP	Stop Condition Setup Time	0.26	—	μs	5	Figure 9 on page 78
t134	Data Hold Time	300	—	ns	4	Figure 9 on page 78
t134SMLFM	Data Hold Time	0	—	ns	4, 5	Figure 9 on page 78
t134SMLFMP	Data Hold Time	0	—	ns	4, 5	Figure 9 on page 78
t135	Data Setup Time	250	—	ns		Figure 9 on page 78
t135SMLFM	Data Setup Time	100	—	ns	5	Figure 9 on page 78
t135SMLFMP	Data Setup Time	50	—	ns	5	Figure 9 on page 78
t136	Device Time Out	25	35	ms	1	
t137	Cumulative Clock Low Extend Time (slave device)	—	25	ms	2	Figure 13 on page 82
<b>continued...</b>						

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t <sub>138</sub>	Cumulative Clock Low Extend Time (master device)	—	10	ms	3	Figure 13 on page 82
T <sub>por</sub>	Time in which a device must be operational after power-on reset	—	500	ms		

Notes: 1. A device will timeout when any clock low exceeds this value.  
2. t<sub>137</sub> is the cumulative time a slave device is allowed to extend the clock cycles in one message from the initial start to stop. If a slave device exceeds this time, it is expected to release both its clock and data lines and reset itself.  
3. t<sub>138</sub> is the cumulative time a master device is allowed to extend its clock cycles within each byte of a message as defined from start-to-ack, ack-to-ack, or ack-to-stop.  
4. t<sub>134</sub> has a minimum timing for I<sup>2</sup>C of 0 ns, while the minimum timing for SMBus/SMLINK is 300 ns.  
5. Timings with the SMLFM designator apply only to SMLink[1,0] when operating in Fast Mode.

**Figure 13. SMBus/SMLink Timeout**

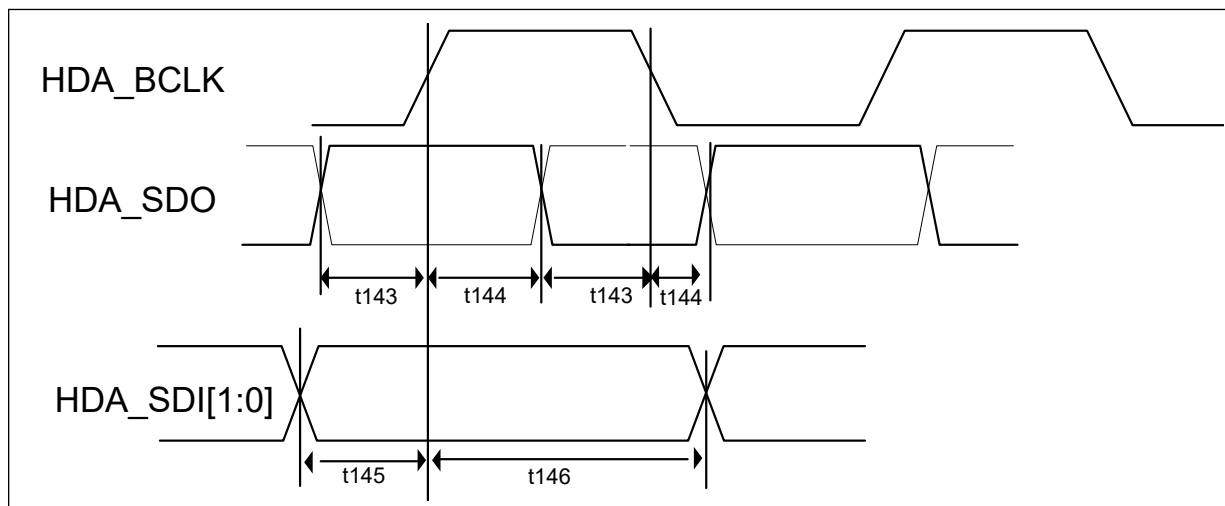


**NOTE**

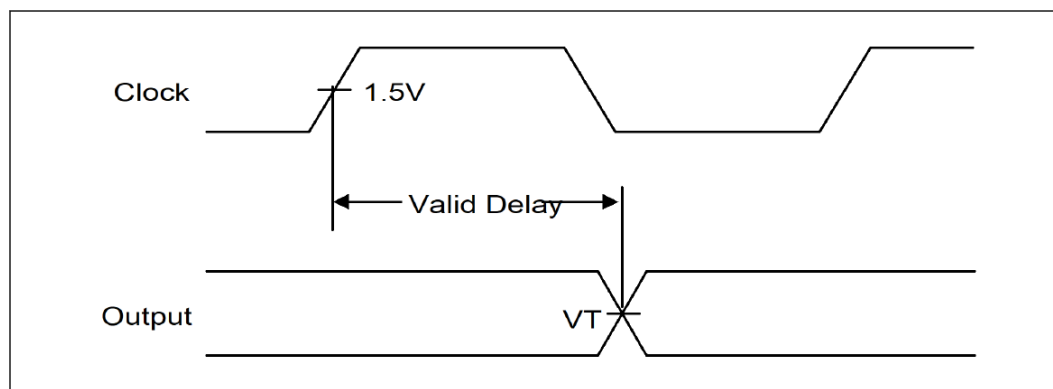
SMBCLK also refers to SML[1:0]CLK and SMBDATA also refers to SML[1:0]DATA in Figure 9 on page 78.

**Table 37. Intel® High Definition Audio (Intel® HD Audio) Timing**

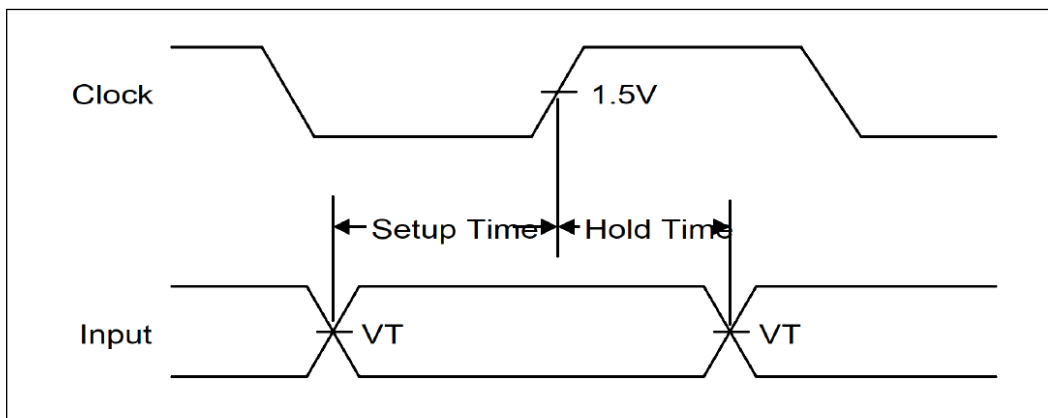
Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t <sub>143</sub>	Time duration for which HDA_SDO is valid before HDA_BCLK edge.	5.5	13.5	ns		Figure 14 on page 83
t <sub>144</sub>	Time duration for which HDA_SDO is valid after HDA_BCLK edge.	5.5	13.5	ns		Figure 14 on page 83
t <sub>145</sub>	Setup time for HDA_SDI[1:0] at rising edge of HDA_BCLK	20	—	ns		Figure 14 on page 83
t <sub>146</sub>	Hold time for HDA_SDI[1:0] at rising edge of HDA_BCLK	3	—	ns		Figure 14 on page 83

**Figure 14. Intel® High Definition Audio (Intel® HD Audio) Input and Output Timings****Table 38. LPC Timing (24 MHz)**

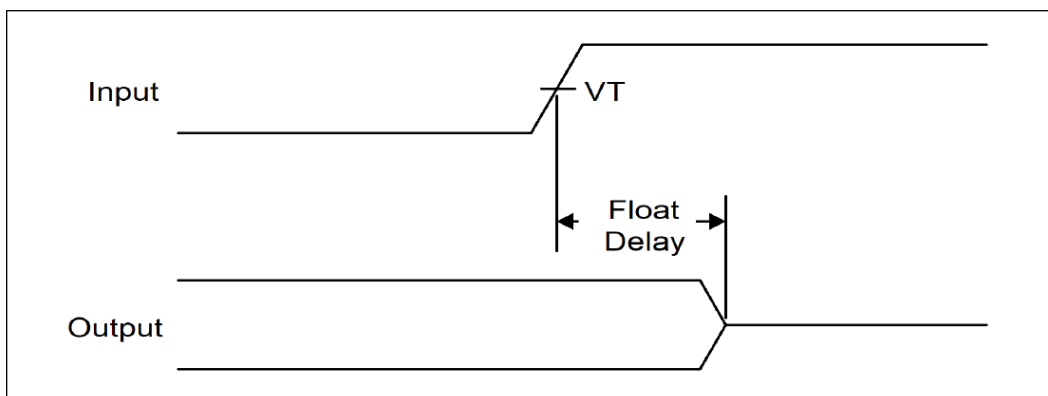
Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t150	LAD[3:0] Valid Delay from CLKOUT_LPC[1:0] Rising	0	23.5	ns		Figure 15 on page 83
t151	LAD[3:0] Output Enable Delay from CLKOUT_LPC[1:0] Rising	2	—	ns		Figure 18 on page 84
t152	LAD[3:0] Float Delay from CLKOUT_LPC[1:0] Rising	—	28	ns		Figure 17 on page 84
t153	LAD[3:0] Setup Time to CLKOUT_LPC[1:0] Rising	20	—	ns		Figure 16 on page 84
t154	LAD[3:0] Hold Time from CLKOUT_LPC[1:0] Rising	2	—	ns		Figure 16 on page 84
t157	LFRAME# Valid Delay from CLKOUT_LPC[1:0] Rising	0	23.5	ns		Figure 15 on page 83

**Figure 15. Valid Delay from Rising Clock Edge**

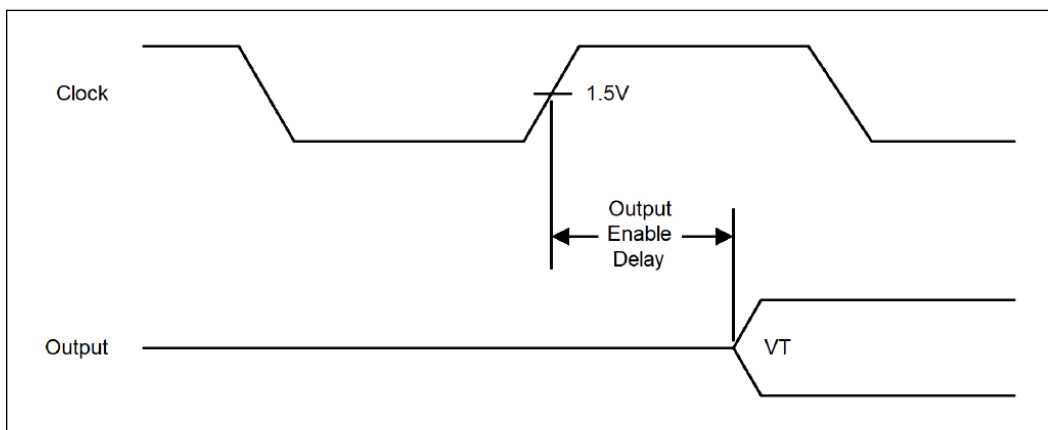
**Figure 16. Setup and Hold Times**



**Figure 17. Float Delay**



**Figure 18. Output Enable Delay**



**Table 39. Miscellaneous Timings**

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t160	SERIRQ Setup Time to PCICLK Rising	7	—	ns		<a href="#">Figure 16 on page 84</a>
t161	SERIRQ Hold Time from PCICLK Rising	0	—	ns		
t162	GPIO, USB Resume Pulse Width	2	—	RTCCLK		<a href="#">Figure 19 on page 85</a>
t163	SPKR Valid Delay from OSC Rising	—	200	ns		<a href="#">Figure 15 on page 83</a>

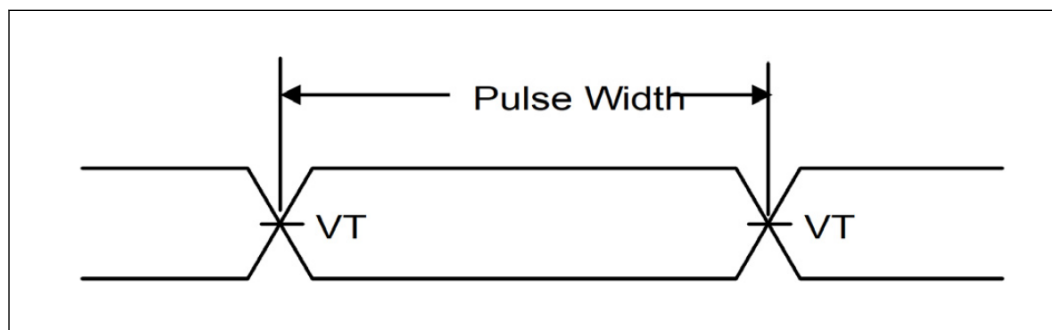
**Figure 19. Pulse Width**

Figure 20. PCH Test Load

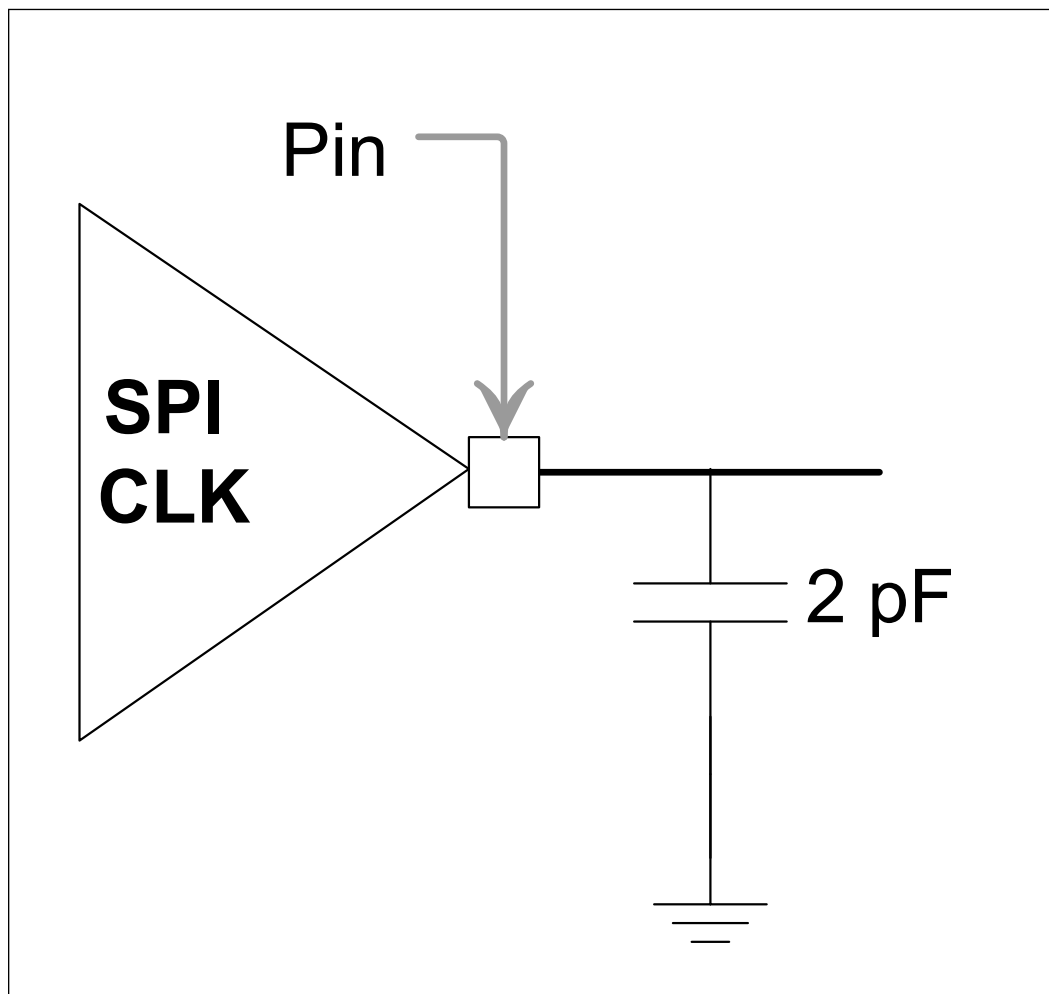


Table 40. SPI Timings (20 MHz)

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t180a	Serial Clock Frequency	16.8	17.48	MHz	1	
t183a	Tco of SPI MOSI and SPI I/O with respect to serial clock falling edge at the host	-13	14	ns		Figure 21 on page 91
t184a	Setup of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	35	—	ns		Figure 21 on page 91
t185a	Hold of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	1.5	—	ns		Figure 21 on page 91
t186a	Setup of SPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		Figure 21 on page 91

continued...

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t187a	Hold of SPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		Figure 21 on page 91
t188a	SPI CLK High time	23.84	—	ns	2	Figure 21 on page 91
t189a	SPI CLK Low time	31.84	—	ns	2	Figure 21 on page 91
Notes: 1. The typical clock frequency driven by the PCH is 17.14 MHz. 2. Measurement point for low time and high time is taken at 0.5(VCCSPI). 3. PCH output timing such as Tco, are simulation values, with a test load of 2pF.						

**Table 41. SPI Timings (33 MHz)**

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t180b	Serial Clock Frequency	29.4	30.6	MHz	1	
t183b	Tco of SPI MOSI and SPI I/O with respect to serial clock falling edge at the host	-8	8	ns		Figure 21 on page 91
t184b	Setup of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	18	—	ns		Figure 21 on page 91
t185b	Hold of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	1.5	—	ns		Figure 21 on page 91
t186b	Setup of SPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		Figure 21 on page 91
t187b	Hold of SPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		Figure 21 on page 91
t188b	SPI CLK High time	16	—	ns	2	Figure 21 on page 91
t189b	SPI CLK Low time	16	—	ns	2	Figure 21 on page 91
Notes: 1. The typical clock frequency driven by the PCH is 30 MHz. 2. Measurement point for low time and high time is taken at 0.5(VCCSPI). 3. PCH output timing such as Tco, are simulation values, with a test load of 2pF.						

**Table 42. SPI Timings (50 MHz)**

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t180c	Serial Clock Frequency	47.04	48.96	MHz	1	
t183c	Tco of SPI MOSI and SPI I/O with respect to serial clock falling edge at the host	-3	6.2	ns		Figure 21 on page 91
t184c	Setup of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	7	—	ns		Figure 21 on page 91
continued...						

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t185c	Hold of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	1.5	—	ns		Figure 21 on page 91
t186c	Setup of SPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		Figure 21 on page 91
t187c	Hold of SPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		Figure 21 on page 91
t188c	SPI CLK High time	7.1	—	ns	2, 3	Figure 21 on page 91
t189c	SPI CLK Low time	11.17	—	ns	2, 3	Figure 21 on page 91
<b>Notes:</b> 1. Typical clock frequency driven by the PCH is 48 MHz. 2. When using 48 MHz mode ensure target flash component can meet t188c and t189c specifications. Measurement should be taken at a point as close as possible to the package pin. 3. Measurement point for low time and high time is taken at 0.5(VCCSPI). 4. PCH output timing such as Tco, are simulation values, with a test load of 2pF.						

**Table 43. SPI Touch Timings (20 MHz)**

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t180d	Serial Clock Frequency	16.8	17.48	MHz	1	
t183d	Tco of SPI MOSI and SPI I/O with respect to serial clock falling edge at the host	-5	10	ns		Figure 21 on page 91
t184d	Setup of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	30	—	ns		Figure 21 on page 91
t185d	Hold of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	1	—	ns		Figure 21 on page 91
t186d	Setup of SPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		Figure 21 on page 91
t187d	Hold of SPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		Figure 21 on page 91
t188d	SPI CLK High time	23.84	—	ns	2, 3	Figure 21 on page 91
t189d	SPI CLK Low time	31.84	—	ns	2, 3	Figure 21 on page 91
<b>Notes:</b> 1. Typical clock frequency driven by the PCH is 17.14 MHz. 2. When using 17 MHz mode ensure target flash component can meet t188d and t189d specifications. Measurement should be taken at a point as close as possible to the package pin. 3. Measurement point for low time and high time is taken at 0.5(VCCSPI). 4. PCH output timing such as Tco, are simulation values, with a test load of 2pF.						

**Table 44. SPI Touch Timings (33 MHz) 1.8 V/3.3 V**

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t180e	Serial Clock Frequency	29.4	30.6	MHz	1	
t183e	Tco of SPI MOSI and SPI I/O with respect to serial clock falling edge at the host	-2.5	7 (3.3 V) 6.5 (1.8 V)	ns		Figure 21 on page 91
t184e	Setup of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	16.5	—	ns		Figure 21 on page 91
t185e	Hold of SPI MISO and SPI I/O with respect to serial clock falling edge at the host	1	—	ns		Figure 21 on page 91
t186e	Setup of SPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		Figure 21 on page 91
t187e	Hold of SPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		Figure 21 on page 91
t188e	SPI CLK High time	16	—	ns	2, 3	Figure 21 on page 91
t189e	SPI CLK Low time	16	—	ns	2, 3	Figure 21 on page 91
<b>Notes:</b> 1. Typical clock frequency driven by the PCH is 30 MHz. 2. When using 30 MHz mode ensure target flash component can meet t188c and t189c specifications. Measurement should be taken at a point as close as possible to the package pin. 3. Measurement point for low time and high time is taken at 0.5(VCCSPI). 4. PCH output timing such as Tco, are simulation values, with a test load of 2pF.						

**Table 45. eSPI Timings (33 MHz)**

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t180f	Serial Clock Frequency	29.4	30.6	MHz	1	
t183f	Tco of eSPI MOSI and eSPI I/O with respect to serial clock falling edge at the host	-7.5	7.5	ns		Figure 21 on page 91
t184f	Setup of eSPI MISO and eSPI I/O with respect to serial clock falling edge at the host	14	—	ns		Figure 21 on page 91
t185f	Hold of eSPI MISO and eSPI I/O with respect to serial clock falling edge at the host	1	—	ns		Figure 21 on page 91
t186f	Setup of eSPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		Figure 21 on page 91
t187f	Hold of eSPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		Figure 21 on page 91
<b>continued...</b>						

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t188f	eSPI CLK High time	16	—	ns	2, 3	Figure 21 on page 91
t189f	eSPI CLK Low time	16	—	ns	2, 3	Figure 21 on page 91
<b>Notes:</b> 1. Typical clock frequency driven by the PCH is 30 MHz. 2. When using 30 MHz mode ensure target flash component can meet t188f and t189f specifications. Measurement should be taken at a point as close as possible to the package pin. 3. Measurement point for low time and high time is taken at 0.5(VCCSPI). 4. PCH output timing such as Tco, are simulation values, with a test load of 2pF.						

**Table 46. eSPI Timings (50 MHz)**

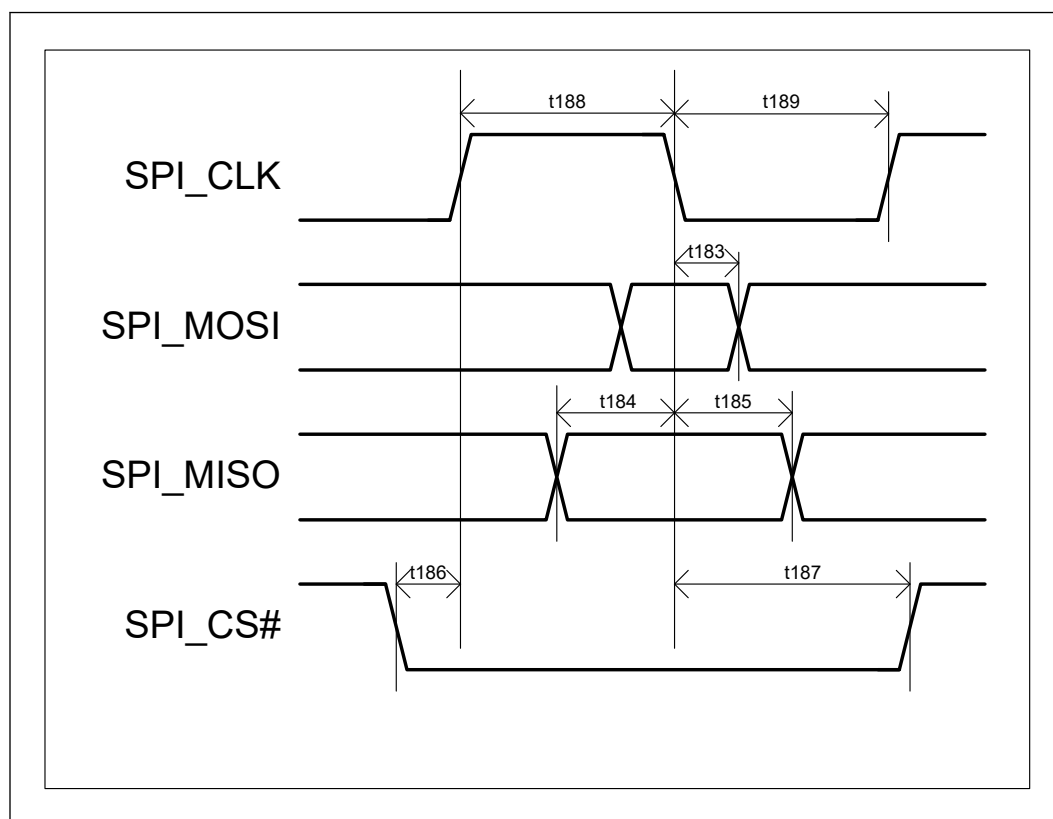
Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t180g	Serial Clock Frequency	47.04	48.96	MHz	1	
t183g	Tco of eSPI MOSI and eSPI I/O with respect to serial clock falling edge at the host	-6	6	ns		Figure 21 on page 91
t184g	Setup of eSPI MISO and eSPI I/O with respect to serial clock falling edge at the host	7.5	—	ns		Figure 21 on page 91
t185g	Hold of eSPI MISO and eSPI I/O with respect to serial clock falling edge at the host	1	—	ns		Figure 21 on page 91
t186g	Setup of eSPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		Figure 21 on page 91
t187g	Hold of eSPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		Figure 21 on page 91
t188g	eSPI CLK High time	7.84	—	ns	2, 3	Figure 21 on page 91
t189g	eSPI CLK Low time	11.84	—	ns	2, 3	Figure 21 on page 91
<b>Notes:</b> 1. Typical clock frequency driven by the PCH is 48 MHz. 2. When using 48 MHz mode ensure target flash component can meet t188g and t189g specifications. Measurement should be taken at a point as close as possible to the package pin. 3. Measurement point for low time and high time is taken at 0.5(VCCSPI). 4. PCH output timing such as Tco, are simulation values, with a test load of 2pF.						

**Table 47. eSPI Timings (66 MHz)**

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t180h	Serial Clock Frequency	58.8	61.2	MHz	1	
t183h	Tco of eSPI MOSI and eSPI I/O with respect to serial clock falling edge at the host	-2.5	1	ns		Figure 21 on page 91
t184h	Setup of eSPI MISO and eSPI I/O with respect to serial clock falling edge at the host	5.5	—	ns		Figure 21 on page 91
t185h	Hold of eSPI MISO and eSPI I/O with respect to serial clock falling edge at the host	1	—	ns		Figure 21 on page 91
<b>continued...</b>						

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t186h	Setup of eSPI CS# assertion with respect to serial clock rising edge at the host	30	—	ns		Figure 21 on page 91
t187h	Hold of eSPI CS# assertion with respect to serial clock falling edge at the host	30	—	ns		Figure 21 on page 91
t188h	eSPI CLK High time	8	—	ns	2, 3	Figure 21 on page 91
t189h	eSPI CLK Low time	8	—	ns	2, 3	Figure 21 on page 91

Notes: 1. Typical clock frequency driven by the PCH is 60 MHz.  
2. When using 60 MHz mode ensure target flash component can meet t188h and t189h specifications. Measurement should be taken at a point as close as possible to the package pin.  
3. Measurement point for low time and high time is taken at 0.5(VCCSPI).  
4. PCH output timing such as Tco, are simulation values, with a test load of 2pF.

**Figure 21. SPI/GSPI/eSPI Timings**

**Table 48. GSPI Timings (25 MHz)**

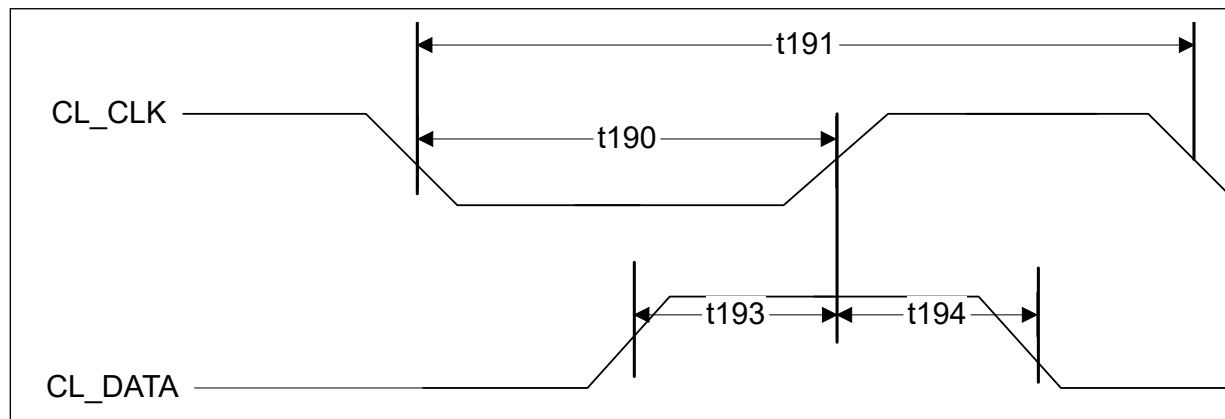
Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
F	Serial Clock Frequency	—	25	MHz		
t183	Tco of SPI MOSI with respect to serial clock falling edge	-15	7.6	ns		Figure 21 on page 91
t184	Setup of SPI MISO and SPI I/O with respect to serial clock rising edge	3.8	—	ns		Figure 21 on page 91
t185	Hold of SPI MISO and SPI I/O with respect to serial clock rising edge	20	—	ns		Figure 21 on page 91
t186	Setup of SPI CS# assertion with respect to serial clock rising edge	20	—	ns		Figure 21 on page 91
t187	Hold of SPI CS# assertion with respect to serial clock falling edge	20	—	ns		Figure 21 on page 91

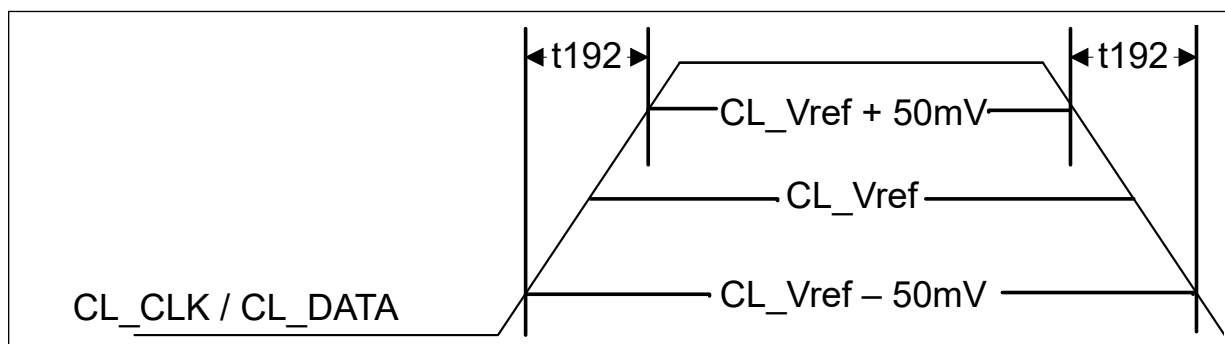
**Table 49. Controller Link Receive Timings**

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
t190	Single bit time	13	—	ns		Figure 22 on page 92
t191	Single clock period	30	—	ns		Figure 22 on page 92
t192	Input Signal Slew Rate	0.02857	0.90909	ns	3	Figure 23 on page 93
t193	Setup time before CL_CLK	0.9	—	ns		Figure 22 on page 92
t194	Hold time after CL_CLK	0.9	—	ns		Figure 22 on page 92
VIL_AC	Input low voltage (AC)	—	CL_Vref - 0.08	V	2	
VIH_AC	Input high voltage (AC)	CL_Vref + 0.08	—	V	2	

Notes: 1. Measured from (CL\_Vref - 50 mV to CL\_Vref + 50 mV) at the receiving device side. No test load is required for this measurement as the receiving device fulfills this purpose.  
2. CL\_Vref = 0.12\*(VCCSus3\_3).

**Figure 22. Controller Link Receive Timings**



**Figure 23. Controller Link Receive Slew Rate****Table 50. UART Timings**

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

**Table 51. I<sup>2</sup>S Timings - Master Mode**

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
<b>SCLK</b>						
F <sub>I2S</sub>	Clock Frequency in (non S0ix)	—	12.288	MHz		
F <sub>I2S</sub>	Clock Frequency (S0ix)	—	9.6	MHz		
<b>SFRM</b>						
TCO	Clock to Output Delay	—8	15	ns		
<b>RXD</b>						
TSU	Setup Time	40	—	ns		
THD	Hold Time	1	—	ns		
<b>TXD</b>						
TCO	Clock to Output Delay	—8	15	ns		

**Table 52. I<sup>2</sup>S Timing - Slave Mode (non S0ix)**

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
<b>SCLK</b>						
F <sub>I2S</sub>	Clock Frequency	—	12.288	MHz		
	Jitter	—	300	ps		
<b>SFRM</b>						
T <sub>SU</sub>	Setup Time	9	—	ns		
T <sub>HD</sub>	Hold Time	10	—	ns		
<b>RXD</b>						
<i>continued...</i>						

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
T <sub>SU</sub>	Setup Time	9	—	ns		
T <sub>HD</sub>	Hold Time	10	—	ns		
<b>TXD</b>						
T <sub>CO</sub>	Clock to Output Delay	0	21	ns		

**Table 53. I<sup>2</sup>S Timing - Slave Mode (S0ix)**

Symbol	Parameter	Minimum	Maximum	Units	Notes	Figure
<b>SCLK</b>						
F <sub>I2S</sub>	Clock Frequency	—	9.6	MHz		
	Jitter	—	300	ps		
<b>SFRM</b>						
T <sub>SU</sub>	Setup Time	15	—	ns		
T <sub>HD</sub>	Hold Time	10	—	ns		
<b>RXD</b>						
T <sub>SU</sub>	Setup Time	15	—	ns		
T <sub>HD</sub>	Hold Time	10	—	ns		
<b>TXD</b>						
T <sub>CO</sub>	Clock to Output Delay	0	28	ns		

## 9.5 Overshoot/Undershoot Guidelines

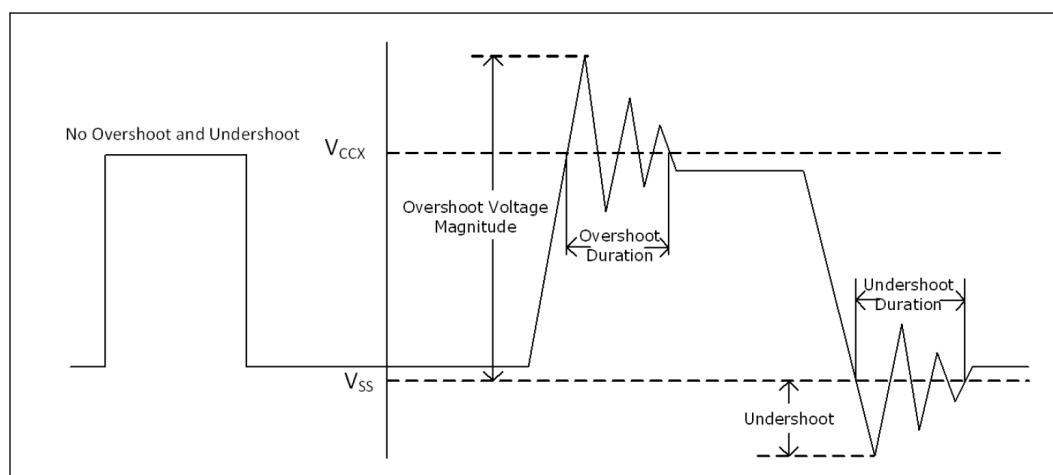
Overshoot (or undershoot) is the absolute value of the maximum voltage above VCC or below VSS. The PCH can be damaged by single and/or repeated overshoot or undershoot events on any input, output, or I/O buffer if the charge is large enough. Baseboard designs that meet signal integrity and timing requirements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

**Table 54. 3.3 V Overshoot/Undershoot Specifications**

Signal Group	Voltage (VCCx) (V)	Overshoot Voltage Magnitude (V)	Overshoot Duration (nS)	Undershoot Voltage Magnitude (V)	Undershoot Duration (nS)
GPP_A, HDA, GPP_G	1.8	2.01	0.6	-0.15	0.6
		1.97	1.2	-0.12	1.2
	3.3	3.65	2.5	-0.26	2.5
		3.61	5	-0.21	5
GPP_F, GPP_E, SPI	1.8	2.01	0.6	-0.15	0.6
		1.97	1.2	-0.12	1.2
	3.3	3.65	2.5	-0.26	2.5
		3.61	5	-0.21	5
continued...					

Signal Group	Voltage (VCCx) (V)	Overshoot Voltage Magnitude (V)	Overshoot Duration (nS)	Undershoot Voltage Magnitude (V)	Undershoot Duration (nS)
GPP_J	1.8	2.27	0.6	-0.42	0.6
		2.2	1.2	-0.34	1.2
GPP_B, GPP_D, GPP_H, GPP_I, GPP_K, GPP_C, GPD	1.8	1.95	0.6	-0.1	0.6
		1.90	1.2	-0.05	1.2
	3.3	3.51	2.5	-0.11	2.5
		3.45	5	-0.05	5
USB 2.0	3.3	4.356	20.825	-1.056	20.835
Notes: 1. These specifications are measured at the PCH pin. 2. Voltage (VCCx) refers to the supply voltage at the pin. Refer below figure for pictorial description of allowable overshoot/undershoot magnitude and duration.					

Figure 24. Maximum Acceptable Overshoot/Undershoot Waveform



## 10.0 Pin Straps

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

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

**Table 55. Pin Straps**

Signal	Usage	When Sampled	Comment
<b>GPP_B14 / SPKR</b>	Top Swap Override	Rising edge of PCH_PWROK	<p>The signal has a weak internal pull-down.  0 = <b>Disable</b> "Top Swap" mode. (Default)  1 = <b>Enable</b> "Top Swap" mode. This inverts an address on access to SPI and firmware hub (FWH), so the processor believes it fetches the alternate boot block instead of the original boot-block. PCH will invert A16 (default) for cycles going to the upper two 64-KB blocks in the FWH or the appropriate address lines (A16, A17 or A18) as selected in Top Swap Block size soft strap.</p> <p><i>Notes:</i> 1. The internal pull-down is disabled after PCH_PWROK is high.  2. Software will not be able to clear the Top Swap bit until the system is rebooted.  3. The status of this strap is readable using the Top Swap bit (Bus 0, Device 31, Function 0, offset DCh, bit4).  4. This signal is in the primary well.</p>
<b>GPP_B18 / GSPI0_MOSI</b>	No Reboot	Rising edge of PCH_PWROK	<p>The signal has a weak internal pull-down.  0 = <b>Disable</b> "No Reboot" mode. (Default)  1 = <b>Enable</b> "No Reboot" mode. PCH will disable the TCO Timer system reboot feature. This function is useful when running ITP/XDP.</p> <p><i>Notes:</i> 1. The internal pull-down is disabled after PCH_PWROK is high.  2. This signal is in the primary well.</p>
<b>GPP_C2 / SMBALERT#</b>	Reserved	Rising edge of RSMRST#	<p>This signal has a weak internal pull-down.</p> <p><i>Notes:</i> 1. The internal pull-down is disabled after RSMRST# de-asserts.  2. This signal is in the primary well.</p>
<b>GPP_B22 / GSPI1_MOSI</b>	Boot BIOS Strap Bit BBS	Rising edge of PCH_PWROK	<p>This Signal has a weak internal pull-down.  This field determines the destination of accesses to the BIOS memory range. Also controllable using Boot BIOS Destination bit (Bus0, Device31, Function0, offset DCh, bit 6).</p>
<i>continued...</i>			

Signal	Usage	When Sampled	Comment	
			Bit 6	Boot BIOS Destination
			0	SPI (Default)
			1	LPC
			Notes: 1. The internal pull-down is disabled after PCH_PWROK is high. 2. If option 1 (LPC) is selected, BIOS may still be placed on LPC, but all platforms are required to have SPI flash connected directly to the PCH's SPI bus with a valid descriptor in order to boot. 3. Boot BIOS Destination select to LPC by functional strap or using Boot BIOS Destination bit will not affect SPI accesses initiated by Intel® CSME or Integrated GbE LAN. 4. This signal is in the primary well.	
<b>GPP_C5 / SML0ALERT#</b>	eSPI or LPC Select	Rising edge of RSMRST#	This signal has a weak internal pull-down. 0 = <b>LPC</b> is selected (for EC). (Default) 1 = <b>eSPI</b> is selected (for EC). Notes: 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well. <b>Warning:</b> If this strap is configured to '0' (eSPI is disabled), the <i>eSPI Flash Sharing Mode</i> strap must be configured to '0' as well (SAFS is disabled)	
<b>SPI0_MOSI</b>	Reserved	Rising edge of RSMRST#	External pull-up is required. Recommend 100 kOhm if pulled up to 3.3 V or 75 kOhm if pulled up to 1.8 V. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.	
<b>GPP_H15 / SML3ALERT#</b>	Reserved	Rising edge of RSMRST#	External pull-up is required. Recommend 100K if pulled up to 3.3 V or 75K if pulled up to 1.8 V. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.	
<b>GPP_B23 / SML1ALERT# / PCHHOT#</b>	Intel® DCI-OOB	Rising edge of RSMRST#	This signal has an internal pull-down. 0 = Disable Intel® DCI-OOB (Default) 1 = Enable Intel® DCI-OOB Notes: 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. When used as PCHHOT# and strap low, a 150 kOhm pull-up is needed to ensure it does not override the internal pull-down strap sampling. 3. This signal is in the primary well.	
<b>SPI0_IO2</b>	Reserved	Rising edge of RSMRST#	External pull-up is required. Recommend 100 kOhm if pulled up to 3.3 V or 75 kOhm if pulled up to 1.8 V. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.	
<b>SPI0_IO3</b>	Reserved	Rising edge of RSMRST#	External pull-up is required. Recommend 100 kOhm if pulled up to 3.3 V or 75 kOhm if pulled up to 1.8 V. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.	

continued...

Signal	Usage	When Sampled	Comment
<b>HDA_SDO / I2S0_TXD</b>	Reserved	Rising edge of PCH_PWROK	This signal has a weak internal pull-down. <i>Notes:</i> 1. The internal pull-down is disabled after PCH_PWROK is high. 2. This signal is in the primary well.
<b>GPP_H12 / SML2ALERT#</b>	eSPI Flash Sharing Mode	Rising edge of RSMRST#	This signal has a weak internal pull-down. 0 = Master Attached Flash Sharing (MAFS) enabled (Default) 1 = Slave Attached Flash Sharing (SAFS) enabled. <i>Notes:</i> 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well. <b>Warning:</b> This strap must be configured to '0' (SAFS is disabled) if the eSPI or LPC strap is configured to '0' (eSPI is disabled)
<b>GPP_I6 / DDPB_CTRLDATA</b>	Display Port B Detected	Rising edge of PCH_PWROK	This signal has a weak internal pull-down. 0 = Port B is not detected. (Default) 1 = Port B is detected. <i>Notes:</i> 1. The internal pull-down is disabled after PCH_PWROK de-asserts. 2. This signal is in the primary well.
<b>GPP_I8 / DDPC_CTRLDATA</b>	Display Port C Detected	Rising edge of PCH_PWROK	This signal has a weak internal Pull-down. 0 = Port C is not detected. (Default) 1 = Port C is detected. <i>Notes:</i> 1. The internal pull-down is disabled after PCH_PWROK is high. 2. This signal is in the primary well.
<b>GPP_I10 / DDPD_CTRLDATA</b>	Display Port D Detected	Rising edge of PCH_PWROK	This signal has a weak internal pull-down. 0 = Port D is not detected. (Default) 1 = Port D is detected. <i>Notes:</i> 1. The internal pull-down is disabled after PCH_PWROK is high. 2. This signal is in the primary well.
<b>ITP_PMODE</b>	Reserved	Rising edge of RSMRST#	This signal has a weak internal pull-up. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.
<b>GPP_F23 / DDPF_CTRLDATA</b>	Display Port F Detected	Rising edge of PCH_PWROK	This signal has a weak internal pull-down. 0 = Port F is not detected. (Default) 1 = Port F is detected. <i>Notes:</i> 1. The internal pull-down is disabled after PCH_PWROK de-asserts. 2. This signal is in the primary well. 3. This strap applies to platforms that support Display Port F only. Refer to the platforms processor documentation for information on Display Port F support.
<b>GPP_J4 / CNV_BRI_DT / UART0B_RTS#</b>	XTAL Frequency Select	Rising edge of RSMRST#	This signal has a weak internal pull-down. An external pull-up is required on this strap since 38.4 MHz XTAL is not supported on the PCH. 0 = 38.4 XTAL frequency selected. (Default; Not supported) 1 = 24 MHz XTAL frequency selected. <i>Notes:</i> 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
<b>continued...</b>			

Signal	Usage	When Sampled	Comment
<b>GPP_J6 / CNV_RGI_DT / UART0B_TXD</b>	M.2 CNV Mode Select	Rising edge of RSMRST#	A weak external pull-up is required. 0 = Integrated CNVi enable. 1 = Integrated CNVi disable. <i>Note:</i> When a RF companion chip is connected to the PCH CNVi interface, the device internal pull-down resistor will pull the strap load to enable CNVi interface.
<b>GPP_J9 / CNV_ MFUART2_TXD</b>	1.8 V VCCSPI	Rising edge of RSMRST#	The signal has a weak internal pull-down 0 = VCCSPI is connected to 3.3 V rail 1 = VCCSPI is connected to 1.8 V rail <i>Note:</i> If VCCSPI is connected to 1.8 V rail, this pin strap must be a '1' for the proper functionality of the SPI (Flash) I/Os.
<b>GPD7</b>	Reserved	Rising edge of DSW_PWROK	External pull-up is required. Recommend 100 kOhm. This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling

## 11.0 8254 Timers

---

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

### Counter 0, System Timer

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

### Counter 2, Speaker Tone

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

## 11.1 Timer Programming

The counter/timers are programmed in the following fashion:

1. Write a control word to select a counter.
2. Write an initial count for that counter.
3. Load the least and/or most significant bytes (as required by Control Word Bits 5, 4) of the 16-bit counter.
4. Repeat with other counters.

Only two conventions need to be observed when programming the counters. First, for each counter, the control word must be written before the initial count is written. Second, the initial count must follow the count format specified in the control word (least significant byte only, most significant byte only, or least significant byte, and then most significant byte).

A new initial count may be written to a counter at any time without affecting the counter's programmed mode. Counting is affected as described in the mode definitions. The new count must follow the programmed count format.

If a counter is programmed to read/write two-byte counts, the following precaution applies – a program must not transfer control between writing the first and second byte to another routine which also writes into that same counter. Otherwise, the counter will be loaded with an incorrect count.

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

- **Control Word Command.** Specifies which counter to read or write, the operating mode, and the count format (binary or BCD).
- **Counter Latch Command.** Latches the current count so that it can be read by the system. The countdown process continues.
- **Read Back Command.** Reads the count value, programmed mode, the current state of the OUT pins, and the state of the Null Count Flag of the selected counter.

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

**Table 56. Counter Operating Modes**

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

## 11.2 Reading From Interval Timer

It is often desirable to read the value of a counter without disturbing the count in progress. There are three methods for reading the counters—a simple read operation, counter Latch command, and the Read-Back command. Each one is explained below:

With the simple read and counter latch command methods, the count must be read according to the programmed format; specifically, if the counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other. Read, write, or programming operations for other counters may be inserted between them.

### Simple Read

The first method is to perform a simple read operation. The counter is selected through Port 40h (Counter 0) or 42h (Counter 2).

---

**NOTE**

Performing a direct read from the counter does not return a determinate value, because the counting process is asynchronous to read operations. However, in the case of Counter 2, the count can be stopped by writing to the GATE bit in Port 61h.

---

**Counter Latch Command**

The Counter Latch command, written to Port 43h, latches the count of a specific counter at the time the command is received. This command is used to ensure that the count read from the counter is accurate, particularly when reading a two-byte count. The count value is then read from each counter's Count register as was programmed by the Control register.

The count is held in the latch until it is read or the counter is reprogrammed. The count is then unlatched. This allows reading the contents of the counters on the fly without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one counter. Counter Latch commands do not affect the programmed mode of the counter in any way.

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

**Read Back Command**

The Read Back command, written to Port 43h, latches the count value, programmed mode, and current states of the OUT pin and Null Count flag of the selected counter or counters. The value of the counter and its status may then be read by I/O access to the counter address.

The Read Back command may be used to latch multiple counter outputs at one time. This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read or reprogrammed. Once read, a counter is unlatched. The other counters remain latched until they are read. If multiple count Read Back commands are issued to the same counter without reading the count, all but the first are ignored.

The Read Back command may additionally be used to latch status information of selected counters. The status of a counter is accessed by a read from that counter's I/O port address. If multiple counter status latch operations are performed without reading the status, all but the first are ignored.

Both count and status of the selected counters may be latched simultaneously. This is functionally the same as issuing two consecutive, separate Read Back commands. If multiple count and/or status Read Back commands are issued to the same counters without any intervening reads, all but the first are ignored.

If both count and status of a counter are latched, the first read operation from that counter returns the latched status, regardless of which was latched first. The next one or two reads, depending on whether the counter is programmed for one or two type counts, returns the latched count. Subsequent reads return unlatched count.

## 12.0 Audio Video and Speech

---

This section will cover some of the details of the separate subsystem parts and how they interact starting with the Intel® HD Audio Controller.

The Intel® HD Audio controller consists of a set of DMA engines that are used to move samples of digitally encoded data between system memory and an internal/external codec(s), with enhancements from the optional Intel® Smart Sound Technology (DSPs). The controller communicates with the internal/external codec(s) over the Audio Link. The Audio Link is a collection of Intel® High Definition Audio serial link, Intel® Display Audio serial link, and other DSP I/O peripheral for proprietary interfaces (For example I<sup>2</sup>S, MIPI\* SoundWire\* and DMIC).

The Intel® HD Audio controller implements a number of output DMA engines and input DMA engines. The Output DMA engines move digital data from system memory to a DAC in a codec. The Input DMA engines move digital data from the ADC in the codec to system memory. The audio link supports a number of internal/external codecs by implementing Serial Data Input signals, one dedicated to each of the supported codecs. Audio software renders outbound, and processes inbound data to/from buffers in memory. The Output DMA engines fetch the digital data from memory and reformat it based on the programmed sample rate, bits/sample and number of channels. The data from the Output DMA engines is then combined and serially sent to the codec(s) over the audio link. The Input DMA engines receive data from the codec(s) over the audio link and format the data based on the programmable attributes for that stream. The data is then written to memory in the predefined format for software to process. Each DMA engine moves one “stream” of data. A single codec can accept or generate multiple “streams” of data, one for each ADC or DAC in the codec. Multiple codecs can accept the same output “stream” processed by a single DMA engine.

The audio subsystem contains serial audio interfaces (I<sup>2</sup>S and Intel® Display Audio Interface), PDM digital microphone interfaces (DMIC), multi-context DMA controllers for transferring data between audio interfaces and memories, on-die SRAMs for storing data and code and serving as a shared L2 memory, and four processor cores (DSPs) based on Cadence\* Tensilica\* HiFi3/LX6 cores.

The Intel® Smart Sound Technology subsystem contains four DSPs are located in their own power domains together with their L1 cache controllers and cache memories. The DSP domains may operate up to 400 MHz. One of the DSPs can be enabled in the D0ix state of the overall audio subsystem, the others are powered down and can be made operational only in D0 state.

The power gating and LDOs are controlled by the Local Power Sequencer (LPS) assisted by the DSP firmware. This approach allows for flexible and robust fine grain power management. The LPS communicates with the Power Management Controller (PMC) to request power supplies.

## Acronyms

Acronyms	Description
DMIC	Digital Microphone. PDM based MEMs microphone modules
DSP	Digital Signal Processor. In AVS specifically an DSP to process audio data.
I <sup>2</sup> S	Inter IC Sound. A serial bus using PCM.
LDO	Low Drop Out. A type of voltage regulator that can regulate output voltage when input voltage is close to output voltage.
MEMs	Micro electrical mechanical Systems. For AVS devices such as Digital MEMs Microphones.
PCM	Pulse Code Modulation. Modulation with amplitude coded into stream.
PDM	Pulse Density Modulation. Modulation with amplitude coded by pulse density.
SoC	System On Chip.
VAD	Voice Activity Detector.
VOIP	Voice Over Internet Protocol
ADC	Analog -to-Digital Converter
DAC	Digital-to-Analog Converter

## 12.1 Signal Description

Name	Type	Description
Intel® High Definition Audio Signals		
HDA_RST#/ I2S1_SCLK	O	Intel® HD Audio Reset: Master H/W reset to internal/external codecs.
HDA_SYNC / I2S0_SFRM	O	Intel® HD Audio Sync:48-kHz fixed rate frame sync to the codecs. Also used to encode the stream number.
HDA_BCLK / I2S0_SCLK	O	Intel® HD Audio Bit Clock:Up to 24 MHz serial data clock generated by the Intel HD Audio controller.
HDA_SDO / I2S0_TXD	O	Intel® HD Audio Serial Data Out: Serial TDM data output to the codecs. The serial output is double-pumped for a bit rate of up to 48 Mb/s.
HDA_SDI0 / I2S0_RXD /	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.
HDA_SDI1 / I2S1_RXD	I/O	
Intel Display Audio Interface		
HDACPU_SCLK	O	Display Audio Bit Clock: Serial data clock generated by the Intel HD Audio controller. PCH supports data rate of up to 96 Mb/s.
HDACPU_SDO	O	Display Audio Serial Data Out: Serial TDM data output to the codec. PCH supports data rate of up to 96 Mb/s.
HDACPU_SDI	I	Display Audio Serial Data In: Serial TDM data input from the codec. PCH supports data rate of up to 96 Mb/s.
I 2S/PCM Interface		
continued..		

Name	Type	Description
<b>I2S0_SCLK</b> / HDA_SDI0	O	<b>I2S/PCM serial bit clock 0:</b> Clock used to control the timing of a transfer. Can be generated internally (Master mode) or taken from an external source (Slave mode).
<b>I2S1_SCLK</b> / HDA_RST	O	<b>I2S/PCM serial bit clock 1:</b> This clock is used to control the timing of a transfer. Can be generated internally (Master mode) or taken from an external source (Slave mode).
<b>I2S2_SCLK</b> / GPP_D8	O	<b>I2S/PCM serial bit clock 2:</b> This clock is used to control the timing of a transfer. Can be generated internally (Master mode) or taken from an external source (Slave mode).
<b>I2S0_SFRM</b> / HDA_SYNC	O	<b>I2S/PCM serial frame indicator 0:</b> This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Master mode) or taken from an external source (Slave mode).
<b>I2S1_SFRM</b> / SNDW2_CLK	O	<b>I2S/PCM serial frame indicator 1:</b> This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Master mode) or taken from an external source (Slave mode).
<b>I2S2_SFRM</b> / GPP_D5 / CNV_RF_RESET#	O	<b>I2S/PCM serial frame indicator 2:</b> This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Master mode) or taken from an external source (Slave mode).
<b>I2S0_TXD</b> / HDA_SDO	O	<b>I2S/PCM transmit data (serial data out)0:</b> This signal transmits serialized data. The sample length is a function of the selected serial data sample size.
<b>I2S1_TXD</b> / SNDW2_DATA	O	<b>I2S/PCM transmit data (serial data out)1:</b> This signal transmits serialized data. The sample length is a function of the selected serial data sample size.
<b>I2S2_TXD</b> / GPP_D6 / MODEM_CLKREQ	O	<b>I2S/PCM transmit data (serial data out)2:</b> This signal transmits serialized data. The sample length is a function of the selected serial data sample size.
<b>I2S0_RXD</b> / HDA_SDI0	I	<b>I2S/PCM receive data (serial data in)0:</b> This signal receives serialized data. The sample length is a function of the selected serial data sample size.
<b>I2S1_RXD</b> / HDA_SDI1	I	<b>I2S/PCM receive data (serial data in)1:</b> This signal receives serialized data. The sample length is a function of the selected serial data sample size.
<b>I2S2_RXD</b> / GPP_D7	I	<b>I2S/PCM receive data (serial data in)2:</b> This signal receives serialized data. The sample length is a function of the selected serial data sample size.
<b>I2S_MCLK</b> / GPP_B11	O	<b>I2S/PCM Master reference clock:</b> This signal is the master reference clock that connects to an audio codec.
<b>DMIC Interface</b>		
<b>DMIC_CLK0</b> / GPP_D19 / SNDW4_CLK	O	<b>Digital Mic Clock:</b> Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz.
<b>DMIC_CLK1</b> / GPP_D17 / SNDW3_CLK	O	<b>Digital Mic Clock:</b> Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz.
<b>DMIC_DATA0</b> / GPP_D20 / SNDW4_DATA	I	<b>Digital Mic Data:</b> Serial data input from the digital mic.
<b>DMIC_DATA1</b> / GPP_D18 / SNDW3_DATA	I	<b>Digital Mic Data:</b> Serial data input from the digital mic.
<b>MIPI* SoundWire* Interface</b>		
<b>SNDW1_CLK</b> / HDA_RST / I2S1_SCLK	O	<b>SoundWire Clock:</b> Serial data clock to external peripheral devices.
<b>SNDW2_CLK</b> / I2S1_SFRM /	O	<b>SoundWire Clock:</b> Serial data clock to external peripheral devices.
<i>continued...</i>		

Name	Type	Description
<b>SNDW3_CLK</b> / GPP_D17/ DMIC_CLK1	O	<b>SoundWire Clock:</b> Serial data clock to external peripheral devices.
<b>SNDW4_CLK</b> / GPP_D19/ DMIC_CLK0	O	<b>SoundWire Clock:</b> Serial data clock to external peripheral devices.
<b>SNDW1_DATA</b> / HDA_SDI1 / I2S1_RXD	I/O	<b>SoundWire Data:</b> Serial data input from external peripheral devices.
<b>SNDW2_DATA</b> / I2S1_TXD	I/O	<b>SoundWire Data:</b> Serial data input from external peripheral devices.
<b>SNDW3_DATA</b> / GPPC_D18 / DMIC_DATA1	I/O	<b>SoundWire Data:</b> Serial data input from external peripheral devices.
<b>SNDW4_DATA</b> / GPPC_D20 / DMIC_DATA0	I/O	<b>SoundWire Data:</b> Serial data input from external peripheral devices.
<b>Misc</b>		
<b>SPKR</b>	O	<b>Speaker Output:</b> Used for connection to external speaker for POST sounds if not using the Intel HD Audio embedded option.

## 12.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value
HDA_SYNC	Pull-down	14 kohm- 26 kohm
HDA_SDO	Pull-down	14 kohm- 26 kohm
HDA_SDI[1:0]	Pull-down	14 kohm- 26 kohm
HDACPU_SDO	Pull-down	14 kohm- 26 kohm
HDACPU_SDI	Pull-down	14 kohm- 26 kohm
I2S[2:0]_SFRM	Pull-down	14 kohm- 26 kohm
I2S[2:0]_SCLK	Pull-down	14 kohm- 26 kohm
I2S[2:0]_TXD	Pull-down	14 kohm- 26 kohm
I2S[2:0]_RXD	Pull-down	14 kohm- 26 kohm
DMIC_DATA[1:0]	Pull-down	14 kohm- 26 kohm
SNDW[4:1]_DATA	Pull-down	14 kohm- 26 kohm
SPKR	Pull-down	14 kohm- 26 kohm

## 12.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>2</sup>	Immediately After Reset <sup>2</sup>	S3/S4/S5	Deep Sx
<b>Intel® High Definition Interface</b>					
HDA_RST#	Primary	Driven Low	Driven Low	Driven Low	OFF
<b>continued...</b>					

Signal Name	Power Plane	During Reset <sup>2</sup>	Immediately After Reset <sup>2</sup>	S3/S4/S5	Deep Sx
HDA_SYNC	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
HDA_BLK	Primary	Driven Low	Driven Low	Driven Low	OFF
HDA_SDO	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
HDA_SDI[1:0]	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
<b>Intel Display Audio Interface</b>					
HDACPU_BCLK	Primary	Driven Low	Driven Low	OFF	OFF
HDACPU_SDO	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
HDACPU_SDI	Primary	Internal Pull-down	Internal Pull-down	OFF	OFF
<b>I<sup>2</sup>S/PCM Interface</b>					
I2S[2:0]_SCLK	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S[2:0]_SFRM	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S0_TXD	Primary	Internal Pull-down	Driven Low	Low then disabled (Refer to note)	OFF
I2S[2:1]_TXD	Primary	Driven Low	Driven Low	Driven Low	OFF
I2S[2:0]_RXD	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S_MCLK	Primary	Driven Low	Driven Low	Driven Low	OFF
<b>DMIC Interface</b>					
DMIC_CLK[1:0]	Primary	Driven Low	Driven Low	Driven Low	OFF
DMIC_DATA[1:0]	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
<b>SoundWire* Interface</b>					
SNDW_DATA[1:4]	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SNDW_CLK[1:4]	Primary	Driven Low	Driven Low	Driven Low	OFF
<b>Misc</b>					
SPKR	Primary	Driven Low	Driven Low	Low then disabled (Refer to note)	OFF
Notes: 1. SPKR and I2S0_SFRM are also straps in which the 20 kOhm pull-down only occurs during the sampling window and then are disabled. 2. Reset reference for primary well pins is RSMRST#.					

## 12.4 AVS Feature Summary

Topics Covered:

- Intel® High Definition Audio Controller Capabilities

- Audio DSP Capabilities
- Intel® High Definition Audio Link Capabilities
- Intel® Display Audio Link Capabilities
- DMIC Interface
- I<sup>2</sup>S/PCM Interface
- SoundWire\* Interface

### **12.4.1 Intel® High Definition Audio Controller Capabilities**

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

### **12.4.2 Audio DSP Capabilities**

- DSP off load for low power audio rendering and recording
- Intel® Wake on Voice supported in Connected Standby and D3/Sx
- High Performance DSP based on Cadence\* Tensilica\* LX6 HIFI3 DSP Cores operating up to 400 MHz
- Various DSP functions provided by DSP Core: MP3, AAC, 3rd Party IP Algorithms, etc.

### 12.4.3 Intel® High Definition Audio Link Capabilities

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

### 12.4.4 Intel® Display Audio Link Capabilities

- One SDI signal to support one display audio codec
- Drives variable frequency (6 MHz to 96 MHz) BCLK to support:
  - SDO single pumped up to 96 Mb/s
  - SDI's single pumped up to 96 Mb/s
- Provides cadence for 44.1 kHz-based sample rate output

### 12.4.5 DMIC Interface

- Two DMIC PDM interfaces with each port capable of supporting up to 4 digital MEMs microphones
- The audio clock output can operate up to 4.8 MHz
- Supports 1.8 V and 3.3 V I/O voltages

### 12.4.6 I<sup>2</sup>S/PCM Interface

- Three bi-directional I<sup>2</sup>S/PCM ports to support up to three I<sup>2</sup>S connections.
- Support of serial bit rates of 19.2 Mbps or 24 Mbps.
- Can support three modes: Slave Mode, Slave Mode with Locally Generated Master Clock, or Master Mode one bi-directional I<sup>2</sup>S / PCM ports to support one I<sup>2</sup>S connection.

### 12.4.7 SoundWire\* Interface

- Four SoundWire\* Interfaces also referred to as ports for connection to platform peripherals.
- Supports only 1.8 V I/O voltage.

## 13.0 Controller Link

The Controller Link interface is used to connect the Intel® CSME to a wireless LAN device supporting Intel® Active Management Technology. The Controller Link interface will transmit data at up to 60 Mbps with the clock frequency at 30 MHz.

### 13.1 Signal Description

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

### 13.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value (Ohm)	Notes
<b>CL_DATA</b>	Pull-up Pull-down	31.25 100	Refer to <a href="#">I/O Signal Planes and States</a> on page 110
<b>CL_CLK</b>	Pull-up Pull-down	31.25 100	

### 13.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>3</sup>	Immediately After Reset <sup>3</sup>	S3/S4/S5	Deep Sx
<b>CL_DATA</b>	Primary	Refer to Notes 1 and 2	Refer to Notes	Internal Pull-down	OFF
<b>CL_CLK</b>	Primary			Internal Pull-down	OFF
<b>CL_RST#</b>	Primary	Driven Low	Driven High	Driven High	OFF

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

### 13.4 External CL\_RST# Pin Driven/Open-drain Mode Support

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

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

## 14.0 Processor Sideband Signals

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

### Acronyms

Acronyms	Description
PECI	Platform Environmental Control Interface

### 14.1 Signal Description

Name	Type	Description
<b>CPUPWRGD</b>	O	Signal to the processor to indicate its primary power is good.
<b>THRMTRIP#</b>	I	Signal from the processor to indicate that a thermal overheating has occurred.
<b>PM_SYNC</b>	O	Power Management Sync: State exchange from the PCH to the Processor
<b>PM_DOWN</b>	I	Power Management Sync: State exchange from the Processor to the PCH
<b>PLTRST_CPU#</b>	O	Platform reset to the Processor
<b>PECI</b>	I/O	Single-wire serial bus for accessing processor digital thermometer
GPP_E3 / <b>CPU_GP0</b>	I	Thermal management signal
GPP_E7 / <b>CPU_GP1</b>	I	
GPP_B3 / <b>CPU_GP2</b>	I	
GPP_B4 / <b>CPU_GP3</b>	I	

### 14.2 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep S
<b>CPUPWRGD</b>	Primary	Undriven	Driven High	OFF	OFF
<b>THRMTRIP#</b>	Primary	Undriven	Undriven	OFF	OFF
<b>PM_SYNC</b>	Primary	Driven Low	Driven Low	OFF	OFF
<b>PM_DOWN</b>	Primary	Undriven	Undriven	OFF	OFF
<b>PLTRST_CPU#</b>	Primary	Driven Low	Driven High	OFF	OFF
<b>PECI</b>	Primary	Undriven	Undriven	OFF	OFF
<b>CPU_GP[3:0]</b>	Primary	Undriven	Undriven	Undriven	OFF

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

## 14.3 Functional Description

- CPUPWRGD signal is output to the processor that indicates that the primary power is ramped up and stable. CPUPWRGD will be undriven by the PCH (high Z) when RSMRST# is asserted and driven high after RSMRST# is de-asserted
- If THRMTRIP# goes active, the processor is indicating an overheat condition, and the PCH will immediately transition to an S5 state. CPU\_GP can be used from external sensors for the thermal management.
- PM\_SYNC is used to provide early warning to the processor that a global reset is in progress and that the memory contents should be saved and placed into self refresh.
- PM\_DOWN is input to PCH indicates the processor wake up event.

## 15.0 Digital Display Signals

### Acronyms

Acronyms	Description
eDP*	embedded Display Port*

### 15.1 Signal Description

Display is divided between processor and PCH.

- The processor houses:
  - Memory interface
  - Display planes
  - Pipes
  - Digital display interfaces/ports
- The PCH houses:
  - DDC bus
  - Hot-Plug Detect
  - Panel Power
  - Backlight controls

The DDC (Display Data Channel) bus is used for communication between the host system and display. A pair of DDC (DDC\_CLK and DDC\_DATA) signals exist on the PCH for each digital port on the processor. DDC is based on I<sup>2</sup>C protocol.

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

The Panel Power and Backlight controls are used to control power for an internal panel and drive the backlight.

**Table 57. Digital Display Signals**

Name	Type	Description
<b>DDPB_HPDP0</b> / GPP_I0 / DISP_MISC0	I	<b>Display Port B:</b> HPD Hot-Plug Detect
<b>DDPC_HPDP1</b> / GPP_I1 / DISP_MISC1	I	<b>Display Port C:</b> HPD Hot-Plug Detect
<b>DDPD_HPDP2</b> / GPP_I2 / DISP_MISC2	I	<b>Display Port D:</b> HPD Hot-Plug Detect
<b>DDPF_HPDP3</b> / GPP_I3 / DISP_MISC3	I	<b>Display Port F:</b> HPD Hot-Plug Detect
<b>DDPB_CTRLCLK</b> / GPP_I5	I/O	<b>Display Port B:</b> Control Clock.
<b>DDPB_CTRLDATA</b> / GPP_I6	I/O	<b>Display Port B:</b> Control Data.
continued...		

Name	Type	Description
<b>DDPC_CTRLCLK</b> / GPP_I7	I/O	<b>Display Port C:</b> Control Clock
<b>DDPC_CTRLDATA</b> / GPP_I8	I/O	<b>Display Port C:</b> Control Data
<b>DDPD_CTRLCLK</b> / GPP_I9	I/O	<b>Display Port D:</b> Control Clock
<b>DDPD_CTRLDATA</b> / GPP_I10	I/O	<b>Display Port D:</b> Control Data
<b>DDPF_CTRLCLK</b> / GPP_F22	I/O	<b>Display Port F:</b> Control Clock
<b>DDPF_CTRLDATA</b> / GPP_F23	I/O	<b>Display Port F:</b> Control Data

## 15.2 Embedded DisplayPort\* (eDP\*) Backlight Control Signals

## 15.3 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
DDPB_CTRLDATA / GPP_I6	Pull-down	15 kOhm-40 kOhm	Refer to the note below.
DDPC_CTRLDATA / GPP_I7	Pull-down	15 kOhm-40 kOhm	Refer to the note below.
<b>DDPD_CTRLDATA</b> / GPP_I10	Pull-down	15 kOhm-40 kOhm	Refer to the note below.
<b>DDPF_CTRLDATA</b> / GPP_F23	Pull-down	15 kOhm-40 kOhm	Refer to the note below.

### NOTE

The internal pull-up/pull-down is only applied during the strap sampling window (PCH\_PWROK) and is then disabled. Enabling can be done using a 2.2 kOhm Pull-up resistor.

## 15.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>DDPB_HPD0</b> /GPP_I0/ DISP_MISC0	Primary	Undriven	Undriven	Undriven	OFF
<b>DDPC_HPD1</b> / GPP_I1/ DISP_MISC1	Primary	Undriven	Undriven	Undriven	OFF
<b>DDPD_HPD2</b> / GPP_I2 / DISP_MISC2	Primary	Undriven	Undriven	Undriven	OFF
<b>DDPF_HPD3</b> /GPP_I3 / DISP_MISC3	Primary	Undriven	Undriven	Undriven	OFF
<b>DDPB_CTRLCLK</b> / GPP_I5	Primary	Undriven	Undriven	Undriven	OFF
<b>DDPB_CTRLDATA</b> / GPP_I6	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
<b>DDPC_CTRLCLK</b> / GPP_I7	Primary	Undriven	Undriven	Undriven	OFF
<b>DDPC_CTRLDATA</b> / GPP_I8	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF

*continued...*

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>DDPD_CTRLCLK</b> / GPP_I9	Primary	Undriven	Undriven	Undriven	OFF
<b>DDPD_CTRLDATA</b> / GPP_I10	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
<b>DDPF_CTRLCLK</b> / GPP_F22	Primary	Undriven	Undriven	Undriven	OFF
<b>DDPF_CTRLDATA</b> / GPP_F23	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
<b>eDP_VDDEN</b> / GPP_F19	Primary	Driven Low	Driven Low	Driven Low	OFF
<b>eDP_BKLTEN</b> / GPP_F20	Primary	Driven Low	Driven Low	Driven Low	OFF
<b>eDP_BKLTCTL</b> / GPP_F21	Primary	Driven Low	Driven Low	Driven Low	OFF
<b>EDP_HPD</b> / GPP_I4 / DISP_MISC4	Primary	Undriven	Undriven	Undriven	OFF
<i>Note:</i> 1. Reset reference for primary well pins is RSMRST#.					

## 16.0 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)

Depending on PCH SKU, the interface can support either one device (via one chip select signal) or two devices (via two chip select signals). Refer to the [PCH SKUs](#) on page 16 for more information. This interface is not shared and distinct from the SPI bus interface used for flash device and TPM.

### NOTE

The PCH LPC and eSPI interfaces are mutually exclusive. A hardware strap is used to determine which interface is used on the platform.

### Acronyms

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

## 16.1 Signal Description

Name	Type	Description
GPP_A9 / CLKOUT_LPC0 / <b>ESPI_CLK</b>	O	<b>eSPI Clock:</b> eSPI clock output from the PCH to slave device.
GPP_A1 / LAD0 / <b>ESPI_IO0</b>	I/O	<b>eSPI Data Signal 0:</b> Bi-directional pin used to transfer data between the PCH and eSPI slave device.
GPP_A2 / LAD1 / <b>ESPI_IO1</b>	I/O	<b>eSPI Data Signal 1:</b> Bi-directional pin used to transfer data between the PCH and eSPI slave device
GPP_A3 / LAD2 / <b>ESPI_IO2</b>	I/O	<b>eSPI Data Signal 2:</b> Bi-directional pin used to transfer data between the PCH and eSPI slave device
GPP_A4 / LAD3 / <b>ESPI_IO3</b>	I/O	<b>eSPI Data Signal 3:</b> Bi-directional pin used to transfer data between the PCH and eSPI slave device
GPP_A5 / LFRAME# / <b>ESPI_CS 0#</b>	O	<b>eSPI Chip Select 0:</b> Driving CS# signal low to select eSPI slave for the transaction.
GPP_A14 / SUS_STAT# / <b>ESPI_RESET#</b>	O	<b>eSPI Reset:</b> Reset signal from the PCH to eSPI slave.

## 16.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value
<b>ESPI_CLK</b>	Pull-down	20 kohm +/- 30%
<b>ESPI_IO[3:0]</b>	Pull-up	20 kohm +/- 30%
<b>ESPI_CS # [1:0]</b>	Pull-up	20 kohm +/- 30%

## 16.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately After Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>ESPI_IO [3:0]</b>	Primary	Internal Pull- up	Internal Pull-down	Internal Pull-down	OFF
<b>ESPI_CLK</b>	Primary	Internal Pull-down	Driven Low	Driven Low	OFF
<b>ESPI_CS [1:0]#</b>	Primary	Internal Pull- up	Driven High	Driven High	OFF
<b>ESPI_ALERT#</b>	Primary	Internal Pull- up	Driven High	Driven High	OFF
<b>ESPI_RESET#</b>	Primary	Driven Low	Driven High	Driven High	OFF

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

## 16.4 Functional Description

Topics Covered:

- Features
- Protocols
- WAIT States from eSPI Slave
- In-Band Link Reset
- Slave Discovery
- Flash Sharing Mode
- Channels and Supported Transactions

### 16.4.1 Features

The PCH eSPI controller supports the following features:

- Support for Master Attached Flash and Slave Attached Flash.
- Support for 20 MHz, 24 MHz, 30 MHz, 48 MHz, and 60 MHz (configured by soft straps).
- 1.8 V support only.
- Up to quad mode support.
- In-band messages for communication between the PCH and slave device to eliminate side-band signals.
- Real time SPI flash sharing, allowing real time operational access by the PCH and slave device.

- Transmitting RTC time/date to the slave device upon request.

**NOTE**

For PCH SKU with two CS#, the PCH eSPI supports two ALERT# pins to support alerts from two supported slave devices to the PCH.

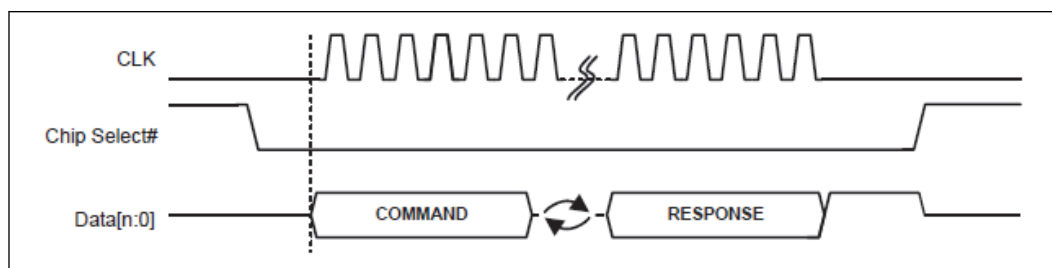
**NOTE**

If only a single slave is connected, the eSPI Compatibility Specification requires that the Slave must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete ALERT# pin.

## 16.4.2 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 [enhanced Serial Peripheral Interface \(eSPI\)](#) on page 117

**Figure 25. Basic eSPI Protocol**



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

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

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

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

## 16.4.3 WAIT States from eSPI Slave

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

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

#### 16.4.4 In-Band Link Reset

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

#### 16.4.5 Slave Discovery

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

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

#### 16.4.6 Flash Sharing Mode

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

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

#### 16.4.7 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 slave is discovered through the GET\_CONFIGURATION command issued by the PCH to the eSPI slave during initialization.

The table below summarizes the eSPI channels and supported transactions.

**Table 58. eSPI Channels and Supported Transactions**

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

#### Peripheral Channel (Channel 0) Overview

The Peripheral channel performs the following functions:

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

### Virtual Wire Channel (Channel 1) Overview

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

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

The table below summarizes the PCH virtual wires in eSPI mode. Refer to the eSPI Compatibility Specification for details.

**Table 59. eSPI Virtual Wires (VW)**

Virtual Wire	PCH Pin Direction	Reset Control	Pin Retained in PCH (For Use by Other Components)
SUS_STAT#	Output	ESPI_RESET#	No
SUSWARN#	Output	ESPI_RESET#	No
SUS_ACK	Input	ESPI_RESET#	No
SUSPWRDNACK	Output	ESPI_RESET#	No
PLTRST#	Output	ESPI_RESET#	Yes
PME# (eSPI Peripheral PME)	Input	ESPI_RESET#	N/A
WAKE#	Input	ESPI_RESET#	No
SMI#	Input	PLTRST#	N/A
SCI#	Input	PLTRST#	N/A
RCIN#	Input	PLTRST#	No
SLP_A#	Output	ESPI_RESET#	Yes
SLP_S3#/SLP_S4#/SLP_S5#/ SLP_LAN#/SLP_WLAN#	Output	DSW_PWROK	Yes
SLAVE_BOOT_LOAD_DONE	Input	ESPI_RESET#	N/A
SLAVE_BOOT_LOAD_STATUS	Input	ESPI_RESET#	N/A
HOST_RST_WARN	Output	PLTRST#	N/A
HOST_RST_ACK	Input	PLTRST	N/A

*continued...*

Virtual Wire	PCH Pin Direction	Reset Control	Pin Retained in PCH (For Use by Other Components)
OOB_RST_WARN	Output	ESPI_RESET#	N/A
OOB_RST_ACK	Input	ESPI_RESET#	N/A
HOST_C10	Output	PLTRST#	N/A
ERROR_NONFATAL	Input	ESPI_RESET#	N/A
ERROR_FATAL	Input	ESPI_RESET#	N/A

### • Interrupt Events

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

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

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

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

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

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

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

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

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

**Figure 26. eSPI Slave Request to PCH for PCH Temperature**

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

**Figure 27. PCH Response to eSPI Slave with PCH Temperature**

PCH to eSPI Slave: Response with PCH Temperature								
Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message = 21h							
1	Tag[3:0]				Length[11:8] = 0h			
2	Length[7:0] = 05h							
3	Dest Slave Addr [7:0] = 0Eh (eSPI slave 0/EC)							0
4	Command Code = 01h (Get_PCH_Temp)							
5	Byte Count = 02h							
6	Source Slave Address [7:1] = 01h (PCH OOB HW Handler)							1
7	PCH Temperature Data [7:0]							

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

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

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

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

**Figure 28. eSPI Slave Request to PCH for PCH RTC Time**

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

**Figure 29. PCH Response to eSPI Slave with RTC Time**

PCH to eSPI Slave: Response with PCH RTC Time								
Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message = 21h							
1	Tag[3:0]				Length[11:8] = 0h			
2	Length[7:0] = 0Ch							
3	Dest Slave Addr [7:0] = 0Eh (eSPI slave 0/EC)							0
4	Command Code = 02h (Get_PCH_RTC_Time)							
5	Byte Count = 09h							
6	Source Slave Address [7:1] = 01h (PCH OOB HW Handler)							1
7	Reserved				DM	HF	DS	
8	PCH RTC Time: Seconds							
9	PCH RTC Time: Minutes							
10	PCH RTC Time: Hours							
11	PCH RTC Time: Day of Week							
12	PCH RTC Time: Day of Month							
13	PCH RTC Time: Month							
14	PCH RTC Time: Year							

**NOTES**

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

**Flash Access Channel (Channel 3) Overview**

**Master Attached Flash (MAF):** This is the configuration where the flash device is directly attached to the PCH and it is the only configuration supported for client platforms. This configuration allows the eSPI device to access the flash device attached to the PCH through a set of flash access commands. These commands are routed to the flash controller and the return data is sent back to the eSPI device.

The Master Attached Flash Channel controller (MAFCC) tunnels flash accesses from eSPI slave to the PCH flash controller. The MAFCC simply provides Flash Cycle Type, Address, Length, Payload (for writes) to the flash controller. The flash controller is

responsible for all the low level flash operations to perform the requested command and provides a return data/status back to the MAFCC, which then tunnels it back to the eSPI slave in a separate completion packet.

The Slave Attached Flash Channel controller (SAFCC) tunnels flash accesses from the PCH to the BMC flash controller. In the case of Slave Attached Flash Sharing (SAFS), the post-security flash request from the SPI Flash Controller (Flash Cycle Type, Address, Length, Payload [for writes]) is routed to the eSPI SAFCC, which tunnels it forward to the BMC over the eSPI link. The completion from the Flash device access, in a separate command, is returned by the BMC to the eSPI SAFCC, which forwards it back to the SPI flash controller.

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

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

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

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

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

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

The Slave will optionally 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 Master.

The SAFCC supports Flash Read, Write and Erase operations. It also supports the RPMC, Read SFDP and Read JEDEC ID commands.

## 17.0 General Purpose Input and Output (GPIO)

The PCH General Purpose Input/Output (GPIO) signals are grouped into multiple groups (GPP\_A, GPP\_B, GPP\_C, GPP\_D, GPP\_E, GPP\_F, GPP\_G, GPP\_H, GPP\_I, GPP\_J, GPP\_K, and GPD group), and 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 (except for GPP\_I group and GPD (3.3 V only), and GPP\_J group (1.8 V only))
- Configurable as an input or output signal.
- Configurable GPIO pad ownership by host, Intel® 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).
- Programmable hardware debouncer (on GPD3/PWRBTN# pin)

**Table 60. GPIO Group Summary**

GPIO Group	Power Pins	Voltage
Primary Well Group A (GPP_A)	VCCPGPPA	1.8 V or 3.3 V
Primary Well Group B (GPP_B) Primary Well Group C (GPP_C)	VCCPGPPBC	1.8 V or 3.3 V
Primary Well Group D (GPP_D)	VCCPGPPD	1.8 V or 3.3 V
Primary Well Group E (GPP_E) Primary Well Group F (GPP_F)	VCCPGPPEF	1.8 V or 3.3 V
Primary Well Group G (GPP_G)	VCCPGPPG_3P3 or VCCPRIM_1P8	1.8 V or 3.3 V
Primary Well Group H (GPP_H) Primary Well Group K (GPP_K)	VCCPGPPHK	1.8 V or 3.3 V
Primary Well Group I (GPP_I)	VCCPRIM_3P3	3.3 V Only
Primary Well Group J (GPP_J)	VCCPRIM_1P8	1.8 V Only
Deep Sleep Well Group (GPD)	VCCDSW_3P3	3.3 V Only
<p><i>Note:</i> Except for GPP_G group, the operating voltage of a GPIO group having voltage configurability (3.3 V or 1.8 V) is selected by both connecting the corresponding power pin and setting the group-voltage-selection soft strap to the desired voltage. GPP_G supports dynamic voltage configuration ONLY when the family is used for SDXC functionality. If the family is used for GPIOs, per pin voltage configuration is NOT supported. GPP_G group voltage is selected by setting the corresponding soft strap only.</p>		

## Acronyms

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

## 17.1 Signal Description

For GPIO pin implementation, including multiplexed native functions, default values, signal states, and other characteristics, refer to GPIO Implementation Summary document.

## 17.2 Functional Description

Topics Covered:

- GPIO Voltage
- Programmable Hardware Debouncer
- Integrated Pull-ups and Pull-downs
- GPPJ\_RCOMP\_1P8 Signal
- Interrupt / IRQ via GPIO Requirement
- SMI#/SCI and NMI
- Timed GPIO
- GPIO Blink (BK) and Serial Blink (SBK)
- GPIO Ownership

### 17.2.1 GPIO Voltage

The following GPIO groups have the per-group voltage configurability. The voltage of all signals in the group is selected by both connecting the corresponding power supply pin and setting the associated soft strap to the desired voltage (3.3 V or 1.8 V), except for GPP\_G group, whose voltage is selected via soft strap only.

- GPP\_A, GPP\_B, GPP\_C, GPP\_D, GPP\_E, GPP\_F, GPP\_G, GPP\_H, and GPP\_K.

The following GPIO groups have fixed operating voltage. The corresponding power pin to these GPIO groups must be connected to the correct voltage:

- GPP\_I and GPD groups: 3.3 V only.
- GPP\_J group: 1.8 V only.

### 17.2.2 Programmable Hardware Debouncer

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

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

### 17.2.3 Integrated Pull-Ups and Pull-Downs

All GPIOs have programmable internal pull-up/pull-down resistors (20 KOhm) which are disabled by default. The internal pull-up/pull-down for each GPIO can be enabled by BIOS programming the corresponding PAD\_CFG\_DW1 register. Refer to Volume 2 (Register Info) for more details. The internal pull-up / pull down can only be implemented if the toggle rate of the GPIO is no more than 300 kHz.

---

#### NOTE

Certain GPIOs used as pin straps have internal PU/PD enabled during reset by default.

---

Refer to [Signal Description](#) on page 128 for information on which GPIO has integrated PU/PD enabled during reset.

### 17.2.4 GPPJ\_RCOMP\_1P8 Signal

The PCH implements the GPPJ\_RCOMP\_1P8 as an external bias resistor to ground. A 200 Ohm (+/- 1%) resistor to ground is required on the signal and this single resistor can be shared with SD\_1P8\_RCOMP and SD\_3P3\_RCOMP on the platform.

### 17.2.5 Interrupt / IRQ via GPIO Requirement

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

### 17.2.6 SMI#/SCI and NMI

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

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

- GPP\_B14, GPP\_B20, GPP\_B23
- GPP\_C[23:22]
- GPP\_D[4:0]
- GPP\_E[8:0]
- GPP\_I[3:0]
- GPP\_G[7:0] (support SMI# only).

### 17.2.7 Timed GPIO

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

Timed GPIO can be an input or an output.

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

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

When Timed GPIO is enabled, the crystal oscillator will not be shut down as crystal clock is needed for the Timed GPIO operation. As a result, SLP\_S0# will not be asserted. This has implication to platform power (such as IDLE or S0ix power). Software should only enable Timed GPIO when needed and disable it when Timed GPIO functionality is not required.

### **17.2.8 GPIO Blink (BK) and Serial Blink (SBK)**

Certain GPIOs are capable of supporting blink and serial blink, indicated as BK and SBK respectively in the GPIO Implementation Summary document (doc# 610942). The BK and SBK are implemented as native functions multiplexed on the selected GPIOs. To enable BK or SBK on a GPIO having the capability, BIOS needs to select the assigned native function for BK or SBK on the GPIO.

### **17.2.9 GPIO Ownership**

Any PCH GPIO can be owned either by the host or the Intel® CSME. If a GPIO is selected and controlled by the Intel® CSME, the GPIO cannot be used by the host anymore.

## 18.0 Intel® Serial I/O Inter-Integrated Circuit (I<sup>2</sup>C) Controllers

The PCH implements four I<sup>2</sup>C controllers for four independent I<sup>2</sup>C interfaces, I2C0-I2C3. Each interface is a two-wire serial interface consisting of a serial data line (SDA) and a serial clock (SCL).

### Acronyms

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

### References

Specification	Location
The I <sup>2</sup> C Bus Specification, Version 6	<a href="http://www.nxp.com/documents/user_manual/UM10204.pdf">www.nxp.com/documents/user_manual/UM10204.pdf</a>

## 18.1 Signal Description

Name	Type	Description
<b>I2C0_SDA</b> / GPP_C16	I/OD	<b>I<sup>2</sup>C Link 0 Serial Data Line</b> External Pull-up resistor is required.
<b>I2C0_SCL</b> / GPP_C17	I/OD	<b>I<sup>2</sup>C Link 0 Serial Clock Line</b> External Pull-up resistor is required.
<b>I2C1_SDA</b> / GPP_C18	I/OD	<b>I<sup>2</sup>C Link 1 Serial Data Line</b> External Pull-up resistor is required.
<b>I2C1_SCL</b> / GPP_C19	I/OD	<b>I<sup>2</sup>C Link 1 Serial Clock Line</b> External Pull-up resistor is required.
<b>I2C2_SDA</b> / GPP_D13/ISH_UART0_RXD	I/OD	<b>I<sup>2</sup>C Link 2 Serial Data Line</b> External Pull-up resistor is required.
<b>I2C2_SCL</b> / GPP_D14/ISH_UART0_TXD	I/OD	<b>I<sup>2</sup>C Link 2 Serial Clock Line</b> External Pull-up resistor is required.
<b>I2C3_SDA</b> / ISH_I2C2_SDA / GPP_D4 / SBK4 / BK4	I/OD	<b>I<sup>2</sup>C Link 3 Serial Data Line</b> External Pull-up resistor is required.
<b>I2C3_SCL</b> /ISH_I2C2_SCL / GPP_D23	I/OD	<b>I<sup>2</sup>C Link 3 Serial Clock Line</b> External Pull-up resistor is required.

## 18.2 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
I2C [3:0]_SDA	Primary	Undriven	Undriven	Undriven	OFF
I2C [3:0]_SCL	Primary	Undriven	Undriven	Undriven	OFF

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

## 18.3 Functional Description

Topics Covered:

- Features
- Protocols Overview
- DMA Controller
- Reset
- Power Management
- Interrupts
- Error Handling
- Programmable SDA Hold Time

### 18.3.1 Features

The I<sup>2</sup>C interfaces support the following features:

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

**NOTES**

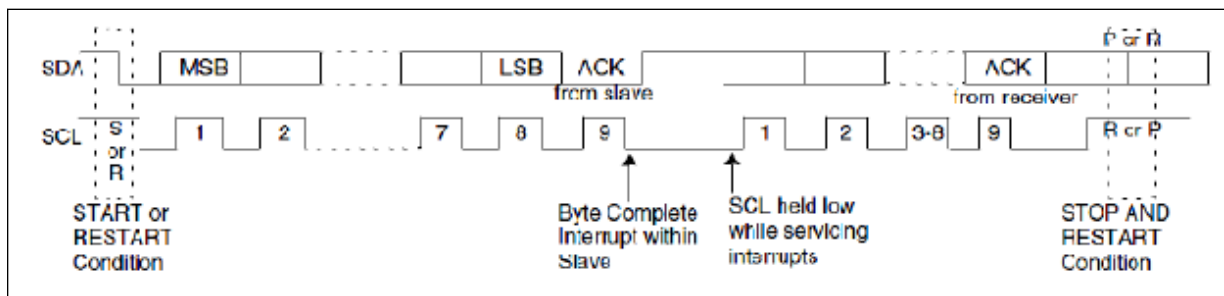
1. The controllers must only be programmed to operate in master mode only. I<sup>2</sup>C slave mode is not supported.
2. I<sup>2</sup>C multi masters is not supported.
3. Simultaneous configuration of Fast Mode and Fast Mode Plus/High speed mode is not supported.
4. I<sup>2</sup>C General Call is not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

**18.3.2 Protocols Overview**

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

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

**Figure 30. Data Transfer on the I<sup>2</sup>C Bus**

**Combined Formats**

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

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

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

### 18.3.3 DMA Controller

The I<sup>2</sup>C 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 be limited by the source or destination transfer widths.
- Address incrementing modes: The DMA has a configurable mechanism for computing the source and destination addresses for the next transfer within the current block. The DMA supports incrementing addresses and constant addresses.
- Flexibility to configure any hardware handshake sideband interface to any of the DMA channels
- Early termination of a transfer on a particular channel.

### 18.3.4 Reset

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

Each host controller and DMA will be in reset state once powered 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<sup>2</sup>C peripheral deadlock condition where the reset goes active in the middle of a transaction, the I<sup>2</sup>C controller must be idle before a reset can be initiated.

---

### 18.3.5 Power Management

#### Device Power Down Support

To power down peripherals connected to PCH I<sup>2</sup>C 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<sup>2</sup>C bus is powered off (power gated). The PCH HW will prevent any transitions on the serial bus signals during a power gate event.

#### Latency Tolerance Reporting (LTR)

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

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

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

### 18.3.6 Interrupts

I<sup>2</sup>C interface has an interrupt line which is used to notify the driver that service is required.

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

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

### **18.3.7 Error Handling**

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

### **18.3.8 Programmable SDA Hold Time**

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

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

### Acronyms

Acronyms	Description
GbE	Gigabit Ethernet

### References

Specification	Location
Alert Standard Format Specification, Version 1.03	<a href="http://www.dmtf.org/standards/asf">http://www.dmtf.org/standards/asf</a>
IEEE 802.3 Fast Ethernet	<a href="https://standards.ieee.org/standard/802_3-2018.html">https://standards.ieee.org/standard/802_3-2018.html</a>
Intel® Ethernet Connection I219 Datasheet	<a href="http://www.intel.com/content/www/us/en/embedded/products/networking/ethernet-connection-i219-datasheet.html">http://www.intel.com/content/www/us/en/embedded/products/networking/ethernet-connection-i219-datasheet.html</a>

## 19.1 Signal Description

**Table 61. GbE LAN Signals**

Name	Type	Description
<b>PCIE5_TXP</b> <b>PCIE5_TXN</b> <b>PCIE9_TXP</b> <b>PCIE9_TXN</b> <b>PCIE12_TXP</b> <b>PCIE12_TXN</b> <b>PCIE13_TXP</b> <b>PCIE13_TXN</b>	O	Refer to <a href="#">PCI Express* (PCIe*)</a> on page 173 for details on the PCI Express transmit signals. <i>Note:</i> For Intel® Ethernet Connection I219 can be connected to one of the following PCI Express ports 5, 9, 12, or 13
<b>PCIE5_RXP</b> <b>PCIE5_RXN</b> <b>PCIE9_RXP</b> <b>PCIE9_RXN</b> <b>PCIE12_RXP</b> <b>PCIE12_RXN</b> <b>PCIE13_RXP</b> <b>PCIE13_RXN</b>	I	Refer to <a href="#">PCI Express* (PCIe*)</a> on page 173 for details on the PCI Express transmit signals. <i>Note:</i> For Intel® Ethernet Connection I219 can be connected to one of the following PCI Express ports 5, 9, 12, or 13
<b>SML0DATA/GPP_C4</b>	I/OD	Refer to <a href="#">System Management Interface and SMLink</a> on page 229 for details on the SML0DATA signal. <i>Note:</i> The Intel® Ethernet Connection I219 connects to SML0DATA signal.
<b>continued...</b>		

Name	Type	Description
<b>SML0CLK</b> /GPP_C3	I/OD	Refer to <a href="#">System Management Interface and SMLink</a> on page 229 for details on the SML0CLK signal. <i>Note:</i> The Intel® Ethernet Connection I219 connects to SML0CLK signal.
<b>LANPHYPC</b> /GPD11	O	<b>LAN PHY Power Control:</b> LANPHYPC should be connected to LAN_DISABLE_N on the PHY. PCH will drive LANPHYPC low to put the PHY into a low power state when functionality is not needed. <i>Note:</i> LANPHYPC can only be driven low if SLP_LAN# is de-asserted.
<b>SLP_LAN#</b>	O	<b>LAN Sub-System Sleep Control:</b> 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. <i>Note:</i> If Gigabit Ethernet Controller is statically disabled via BIOS, SLP_LAN# will be driven low.
<b>LAN_WAKE#</b> /GPD2	I	<b>LAN WAKE:</b> LAN Wake Indicator from the GbE PHY. <i>Note:</i> LAN_WAKE# functionality is only supported with Intel PHY I219. Connection of a third party LAN device's wake signal to LAN_WAKE# is not supported.

## 19.2 Integrated Pull-Ups and Pull-Downs

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

## 19.3 I/O Signal Planes and States

**Table 62. Power Plane and States for Output Signals**

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

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

**Table 63. Power Plane and States for Input Signals**

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

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

The Intel® Ethernet Connection I219 only runs at a speed of 1250 Mbps, which is 1/2 of the 2.5 Gb/s PCI Express frequency. Each of the PCIe 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 PCIe compliant.

---

**NOTE**

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

---

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

---

**NOTE**

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

---

The integrated GbE controller provides a system interface using a PCIe 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 IEEE 802.3, IEEE 802.3u, IEEE 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
  - PCIe/SMLink interface to GbE PHYs
- **Host Interface Features**
  - 64 bit address master support for systems using more than 4 GB of physical memory
  - Programmable host memory receive buffers (256 bytes to 16 KB)

- Intelligent interrupt generation features to enhance driver performance
- Descriptor ring management hardware for transmit and receive
- Software controlled reset (resets everything except the configuration space)
- Message Signaled Interrupts
- **Performance Features**
  - Configurable receive and transmit data FIFO, programmable in 1 KB increments
  - TCP segmentation off loading features
  - Fragmented UDP checksum off load for packet reassembly
  - IPv4 and IPv6 checksum off load support (receive, transmit, and large send)
  - Split header support to eliminate payload copy from user space to host space
  - Receive Side Scaling (RSS) with two hardware receive queues
  - Supports 9018 bytes of jumbo packets
  - Packet buffer size 32 KB
  - TimeSync off load compliant with IEEE 802.1as specification
  - Platform time synchronization
- **Power Management Features**
  - Magic Packet\* wake-up enable with unique MAC address
  - ACPI register set and power down functionality supporting D0 and D3 states
  - Full wake up support (APM, ACPI)
  - MAC power down at Sx, DM-Off with and without WoL
  - Auto connect battery saver at S0 no link and Sx no link
  - Energy Efficient Ethernet (EEE) support
  - Latency Tolerance Reporting (LTR)
  - ARP and ND proxy support through LAN Connected Device proxy
  - Wake on LAN (WoL) from Deep Sx
  - Windows\* InstantGo\* Support

### 19.4.1 GbE PCI Express\* Bus Interface

The GbE controller has a PCIe\* interface to the host processor and host memory. The following sections detail the bus transactions.

#### Transaction Layer

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

#### Data Alignment

- **4 KB Boundary**

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

- **PCI Request Size**

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

#### **Configuration Request Retry Status**

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

## **19.4.2 Error Events and Error Reporting**

### **Completer Abort Error Handling**

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

### **Unsupported Request Error Handling**

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

## **19.4.3 Ethernet Interface**

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

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

## Intel® Ethernet Connection I219

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

The integrated GbE controller supports various modes as listed in the table below.

**Table 64. LAN Mode Support**

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

### 19.4.4 PCI Power Management

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

#### NOTE

The Intel® Ethernet Connection I219 must be powered during the Deep Sx state in order to support host wake up from Deep Sx. GPD\_2\_LAN\_WAKE# on the PCH must be configured to support wake from Deep Sx and must be connected to LANWAKE\_N on the Platform LAN Connect Device. The SLP\_LAN# signal must be driven high (de-asserted) in the Deep Sx state to maintain power to the Platform LAN Connect Device.

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

#### NOTE

SLP\_LAN# pin behavior are detailed in [SLP\\_LAN# Pin Behavior](#).

## 20.0 Interrupt Interface

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

### Acronyms

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

### 20.1 Signal Description

Name	Type	Description
GPP_A6/ <b>SERIRQ</b> / ESPI_CS1#	I/O	Serial Interrupt Request <i>Note:</i> An external Pull-up is required
GPP_A7 / <b>PIRQA#</b> / ESPI_ALERT0#	I/OD	PCI Interrupt Request Ax <i>Note:</i> An external Pull-up is required

### 20.2 Integrated Pull-Ups and Pull-Downs

None.

### 20.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>SERIRQ</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>PIRQA#</b>	Primary	Undriven	Undriven	Undriven	OFF

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

### 20.4 Functional Description

The PCH supports both APIC and PIC modes.

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

- Shareable interrupts requires the Interrupt Controller to track the Assert/De-assert Sideband message from each interrupt source. The Interrupt Controller achieves this through Source ID decode of the message.

- Maintains backwards compatibility with the prior generations where only the lower 24 IRQs are available to support Interrupt Sharing.
- Interrupts are dedicated and not shareable from the perspective of the Interrupt Controller for IRQ 24-119. In other words, not more than 1 Interrupt Initiator is allowed to be assigned to the same IRQ# for IRQ 24-119. For example, GPIO (multi-cause Interrupt Initiator) and Intel® Serial I/O interfaces (I<sup>2</sup>C, UART, GSPI) (multi-function Interrupt Initiator) should not both generate Assert/De-assert IRQn that maps to IRQ24.
- Possible multi-cause Interrupt Initiator that maps to IRQ24-119 are GPIO, eSPI, and so on.
- Possible multi-function Interrupt Initiators that maps to IRQ24-119 are HD Audio, I<sup>2</sup>C/UART/GSPI (Intel Serial I/O Interfaces), ISH, and so on.

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

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

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

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

**Table 65. Interrupt Options - 8259 Mode**

IRQ#	Pin	SERIRQ	PCI Message	Internal Modules
0	No	No	No	8254 Counter 0, HPET#0
1	No	Yes	No	Option for configurable sources including GPIO, eSPI and internal PCI/ACPI devices
2	No	No	No	8259 #2 cascade only
3:7	PIRQA	Yes	Yes	Option for configurable sources including PIRQx, GPIO, eSPI and internal PCI/ACPI devices
8	No	No	No	RTC, HPET#1
9:10	PIRQA	Yes	Yes	Option for configurable sources including PIRQx, GPIO, eSPI, internal PCI/ACPI devices, SCI and TCO.
11	PIRQA	Yes	Yes	Option for configurable sources including PIRQx, GPIO, eSPI, internal ACPI devices, SCI, TCO, HPET #2
<i>continued...</i>				

IRQ#	Pin	SERIRQ	PCI Message	Internal Modules
12	PIRQA	Yes	Yes	Option for configurable sources including PIRQx, GPIO, eSPI, internal ACPI devices, HPET#3
13	No	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI devices
14:15	PIRQA	Yes	Yes	Option for configurable sources including PIRQx, GPIO, eSPI and internal ACPI devices
<b>Notes:</b> 1. 8259 Interrupt Request Lines 0, 2 and 8 are non-shareable and dedicated. Only one interrupt source is allowed to use the Interrupt Request Line at any one time. 2. If an interrupt is used for PCI IRQ [A:H], SCI, or TCO, it should not be used for ISA-style interrupts (via SERIRQ). 3. In 8259 mode, PCI interrupts are mapped to IRQ3, 4, 5, 6, 7, 9, 10, 11, 12, 14, or 15. It can be programmed via PIRQ[A-H] Routing Control at ITSS Private CR + Offset 3100h-3107h.				

**Table 66. Interrupt Options - APIC Mode**

IRQ#	Pin	SERIRQ	PCI Message	IRQ Sharable?	Internal Modules
0	No	No	No	No	Cascade from 8259 #1
1	No	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/PCI devices
2	No	No	No	No	8254 Counter 0, HPET #0 (legacy mode)
3:7	No	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/PCI devices
8	No	No	No	No	RTC, HPET #1 (legacy mode)
9:10	No	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/PCI devices, SCI and TCO
11	No	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/PCI devices, SCI, TCO, HPET #2
12	No	Yes	No	Yes	Option for configurable sources including GPIO, eSPI, internal ACPI/PCI devices, HPET#3
13	No	No	No	Yes	Option for configurable sources including GPIO, eSPI and internal ACPI/PCI devices
14:15	No	Yes	No	Yes	Option for configurable sources including GPIO, eSPI and internal ACPI/PCI devices
16	PIRQA	PIRQA	Yes	Yes	Option for configurable sources including internal PIRQA, GPIO, eSPI and internal ACPI/PCI devices
17:19	No	PIRQ[B-D]	Yes	Yes	Option for configurable sources including internal PIRQ[B-D], GPIO, eSPI and internal ACPI/PCI devices
<b>continued...</b>					

IRQ#	Pin	SERIRQ	PCI Message	IRQ Sharable?	Internal Modules
20:23	No	No	No	Yes	Option for configurable sources including internal PIRQ[E-H], GPIO, eSPI, SCI, TCO, internal ACPI/PCI devices and HPET
24:119	No	No	No	No	Option for configurable sources including GPIO, eSPI and internal ACPI/PCI devices
<p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>1. Interrupts 24 through 119 are dedicated and not shareable from the perspective of the Interrupt Controller. Not more than 1 Interrupt source is allowed to be assigned to the same IRQ#. For example, GPIO and Intel® Serial I/O interfaces (I<sup>2</sup>C, UART, GSPI) should not generate Assert/Deassert_IRQn that maps to IRQ24. Although dedicated, Interrupts 24 through 119 can be configured to be level or edge-triggered.</li> <li>2. If an interrupt is used for PCI IRQ [A:H], SCI, or TCO, it should not be used for ISA-style interrupts (via SERIRQ).</li> <li>3. In APIC mode, the PCI interrupts [A:H] are directly mapped to IRQ[16:23].</li> <li>4. When programming the polarity of internal interrupt sources on the APIC, interrupts 0 through 15, and 24 through 119 receive active-high internal interrupt sources; interrupts 16 through 23 receive active-low internal interrupt sources.</li> <li>5. The internal ACPI/PCI devices refer to PCI/PCIe devices configured to the ACPI or PCI function mode. If in ACPI function mode, the device interrupt is map directly to one of the available IRQ. If in PCI function mode, the device interrupt is map to INT[A-D] and then to the IRQ before these devices issue the Interrupt Message using Assert/Deassert_IRQn.</li> <li>6. PCI Message refers to the downstream Assert/Deassert_INT[A-D] messages forwarded from the processor complex.</li> </ol>					

**Table 67. Interrupt Logic Signals**

Signal Name	C3	S3	S5
SERIRQ	Can be running	OFF	OFF
PIRQA#	Can go active	OFF	OFF

## 20.4.1 8259 Interrupt Controllers (PIC)

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

**Table 68. Interrupt Controllers PIC**

8259	8259 Input	Typical Interrupt Source	Connected Pin/Function
Master	0	Internal	Internal Timer/Counter 0 output or Multimedia Timer #0
	1	Keyboard	IRQ1 via SERIRQ. Option for configurable sources including eSPI, GPIO, internal ACPI devices.
	2	Internal	Slave Controller INTR output
	3	Serial Port A	IRQ3 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.
	4	Serial Port B	IRQ4 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.
	5	Parallel Port/Generic	IRQ5 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.
	6	Floppy Disk	IRQ6 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.
	7	Parallel Port/Generic	IRQ7 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.
<b>continued...</b>			

8259	8259 Input	Typical Interrupt Source	Connected Pin/Function
Slave	0	Real Time Clock	Inverted IRQ8# from internal RTC or Multimedia Timer #1
	1	Generic	IRQ9 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices, SCI, TCO.
	2	Generic	IRQ10 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices, SCI, TCO.
	3	Generic	IRQ11 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices, SCI, TCO or HPET #2.
	4	PS/2 Mouse	IRQ12 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices, SCI, TCO or HPET #3.
	5	Internal	IRQ13 from configurable sources including PIRQx, eSPI, GPIO, internal ACPI devices.
	6	Internal	IRQ14 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.
	7	Internal	IRQ15 from configurable sources including PIRQx, SERIRQ, eSPI, GPIO, internal ACPI devices.

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

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

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

## 20.4.2 Interrupt Handling

### Generating Interrupts

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

**Table 69. Interrupt Status Registers**

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

### Acknowledging Interrupts

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

**Table 70. Content of Interrupt Vector Byte**

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

### Hardware/Software Interrupt Sequence

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

### 20.4.3 Initialization Command Words (ICWx)

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

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

#### ICW1

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

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

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

#### ICW2

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

#### ICW3

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

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

#### ICW4

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

#### 20.4.4 Operation Command Words (OCW)

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

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

#### 20.4.5 Modes of Operation

##### Fully-Nested Mode

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

##### Special Fully-Nested Mode

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

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

##### Automatic Rotation Mode (Equal Priority Devices)

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

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

### Specific Rotation Mode (Specific Priority)

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

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

### Poll Mode

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

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

### Edge and Level Triggered Mode

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

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

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

### End Of Interrupt (EOI) Operations

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

### Normal End of Interrupt

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

### Automatic End of Interrupt Mode

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

## 20.4.6 Masking Interrupts

### Masking on an Individual Interrupt Request

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

### Special Mask Mode

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

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

## 20.4.7 Steering PCI Interrupts

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

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

## 20.5 Advanced Programmable Interrupt Controller (APIC) (D31:F0)

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

### 20.5.1 Interrupt Handling

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

- **Method of Interrupt Transmission:** The I/O APIC transmits interrupts through memory writes on the normal data path to the processor, and interrupts are handled without the need for the processor to run an interrupt acknowledge cycle.
- **Interrupt Priority:** The priority of interrupts in the I/O APIC is independent of the interrupt number. For example, interrupt 10 can be given a higher priority than interrupt 3.
- **More Interrupts:** The I/O APIC in the PCH supports a total of 24 interrupts.
- **Multiple Interrupt Controllers:** The I/O APIC architecture allows for multiple I/O APIC devices in the system with their own interrupt vectors.

### 20.5.2 Interrupt Mapping

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

### 20.5.3 PCI/PCI Express\* Message-Based Interrupts

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

### 20.5.4 IOxAPIC Address Remapping

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

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

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

### 20.5.5 External Interrupt Controller Support

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

## 20.6 Serial Interrupt

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

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

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

---

#### NOTE

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

---

### 20.6.1 Start Frame

The serial IRQ protocol has two modes of operation which affect the start frame. These two modes are:

- Continuous, where the PCH is solely responsible for generating the start frame
- Quiet, where a serial IRQ peripheral is responsible for beginning the start frame.

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

When the serial IRQ stream enters quiet mode (signaled in the Stop Frame), the SERIRQ line remains inactive and pulled up between the Stop and Start Frame until a peripheral drives the SERIRQ signal low. The PCH senses the line low and continues to drive it low for the remainder of the Start Frame. Since the first PCI clock of the start

frame was driven by the peripheral in this mode, the PCH drives the SERIRQ line low for 1 PCI clock less than in continuous mode. This mode of operation allows for a quiet, and therefore lower power, operation. Data Frames.

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

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

## 20.6.2 Stop Frame

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

**Table 71. Stop Frame Explanation**

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

## 20.6.3 Specific Interrupts Not Supported Using SERIRQ

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

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

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

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

**Table 72. Data Frame Format**

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

## 21.0 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<sup>2</sup>C, 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

### Acronyms

Acronyms	Description
Intel® CSME	Intel® Converged Security and Management Engine
I <sup>2</sup> C	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

## References

Specification	Location
I <sup>2</sup> C Specification Version 6.0	<a href="http://www.nxp.com/docs/en/user-guide/UM10204.pdf">http://www.nxp.com/docs/en/user-guide/UM10204.pdf</a>

## 21.1 Signal Description

Name	Type	Description
<b>ISH_I2C0_SDA</b> / GPP_H19	I/OD	ISH I <sup>2</sup> C 0 Data
<b>ISH_I2C0_SCL</b> / GPP_H20	I/OD	ISH I <sup>2</sup> C 0 Clk
<b>ISH_I2C1_SDA</b> / GPP_H21	I/OD	ISH I <sup>2</sup> C 1 Data
<b>ISH_I2C1_SCL</b> / GPP_H22	I/OD	ISH I <sup>2</sup> C 1 Clk
<b>ISH_I2C2_SDA</b> / GPP_D4 / I2C3_SDA / SBK4 / BK4	I/OD	ISH I <sup>2</sup> C 2 Data
<b>ISH_I2C2_SCL</b> / GPP_D23 / I2C3_SCL	I/OD	ISH I <sup>2</sup> C 2 Clk
<b>ISH_GP0</b> /GPP_A18	I/O	ISH GPIO 0
<b>ISH_GP1</b> /GPP_A19	I/O	ISH GPIO 1
<b>ISH_GP2</b> /GPP_A20	I/O	ISH GPIO 2
<b>ISH_GP3</b> /GPP_A21	I/O	ISH GPIO 3
<b>ISH_GP4</b> /GPP_A22	I/O	ISH GPIO 4
<b>ISH_GP5</b> /GPP_A23	I/O	ISH GPIO 5
<b>ISH_GP6</b> / GPP_A12 / BM_BUSY# / SX_EXIT_HOLDOFF#	I/O	ISH GPIO 6
<b>ISH_GP7</b> / GPP_A17 / SD_VDD1_PWR_EN#	I/O	ISH GPIO 7
<b>ISH_UART0_TXD</b> / GPP_D14 / I2C2_SCL	O	ISH UART 0 Transmit Data
<b>ISH_UART0_RXD</b> /GPP_D13 / I2C2_SDA	I	ISH UART 0 Receive Data
<b>ISH_UART0_RTS#</b> /GPP_D15 / GSPi2_CS1# / CNV_WFEN	O	ISH UART 0 Request To Send
<b>ISH_UART0_CTS#</b> /GPP_D16 / CNV_WCEN	I	ISH UART 0 Clear to Send
<b>ISH_UART1_TXD</b> /GPP_C13 / UART1_TXD	O	ISH UART 1 Transmit Data
<b>ISH_UART1_RXD</b> /GPP_C12 / UART1_RXD	I	ISH UART 1 Receive Data
<b>ISH_UART1_RTS#</b> /GPP_C14 / UART1_RTS#	O	ISH UART 1 Request To Send
<b>ISH_UART1_CTS#</b> / GPP_C15 / UART1_CTS#	I	ISH UART 1 Clear to Send
<b>ISH_SPI_CS#</b> / GPP_D9 / GSPi2_CS0#	O	ISH SPI 2 Chip Select
<i>continued...</i>		

Name	Type	Description
<b>ISH_SPI_CLK</b> / GPP_D10 / GSPI2_CLK	O	ISH SPI 2 Clock
<b>ISH_SPI_MISO</b> / GPP_D11 / GSPI2_MISO / GP_BSSB_CLK	I	ISH SPI 2 MISO
<b>ISH_SPI_MOSI</b> / GPP_D12 / GSPI2_MOSI / GP_BSSB_DI	O	ISH SPI 2 MOSI

## 21.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value
<b>ISH_SPI_MISO</b>	Pull-Down	20 kOhm $\pm$ 30%

## 21.3 I/O Signal Planes and States

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

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

## 21.4 Functional Description

Topics Covered:

- ISH Micro-Controller

- SRAM
- PCI Host Interface
- Power Domains and Management
- ISH IPC
- ISH Interrupt Handling via IOAPIC (Interrupt Controller)
- ISH I<sup>2</sup>C Controllers
- ISH UART Controller
- ISH GSPI Controller
- ISH GPIOs

### **21.4.1 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.

### **21.4.2 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 kbytes organized into banks of 32 kB each and is 32-bit wide. The SRAM is shared with Intel<sup>®</sup> 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.

### **21.4.3 PCI Host Interface**

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

#### **MMIO Space**

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

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

#### **DMA Controller**

The DMA controller supports up to 64-bit addressing.

#### **PCI Interrupts**

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

### PCI Power Management

PME is not supported in ISH.

## 21.4.4 Power Domains and Management

### ISH Power Management

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

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

### External Sensor Power Management

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

## 21.4.5 ISH IPC

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

**Table 73. IPC Initiator -> Target Flows**

Initiator	Target
ISH	Host processor
Host processor	ISH
ISH	Intel <sup>®</sup> CSME
Intel <sup>®</sup> 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.

## 21.4.6 ISH Interrupt Handling via IOAPIC (Interrupt Controller)

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

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

### 21.4.7 ISH I<sup>2</sup>C Controllers

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

The ISH's I<sup>2</sup>C host controllers share the same general specifications:

- Master Mode Only (all peripherals must be slave devices)
- Support for the following operating speeds:
  - Standard mode: 100 kbps
  - Fast Mode: 400 kbps
  - Fast Mode Plus: 1000 kbps
  - High Speed Mode: 2400 kbps
- Support for both 7-bit and 10-bit addressing formats on the I<sup>2</sup>C bus
- FIFO of 64 bytes with programmable watermarks/thresholds

### 21.4.8 ISH UART Controller

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

The UART has the following Capabilities:

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

### 21.4.9 ISH GSPI Controller

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

The SPI controller includes:

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

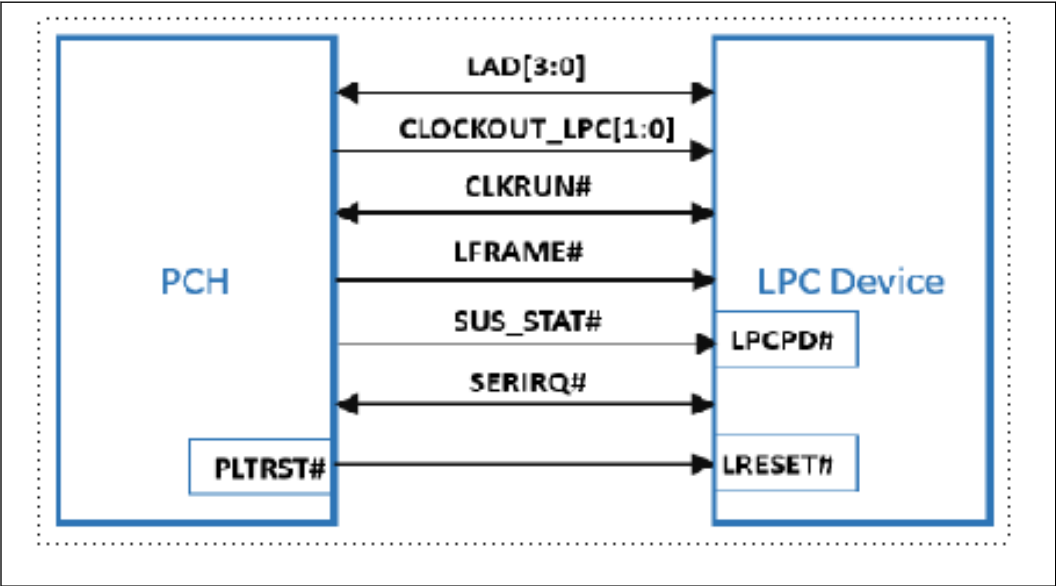
### 21.4.10 ISH GPIOs

The ISH supports eight dedicated GPIOs. Refer to [Signal Description](#) on page 158 for ISH GPIO pins.

## 22.0 Low Pin Count (LPC)

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

**Figure 31. LPC Interface Diagram**



### Acronyms

Acronyms	Description
LPC	Low Pin Count

### References

Specification	Location
Intel® Low Pin Count Interface Specification Revision 1.1	<a href="https://www.intel.com/content/dam/www/program/design/us/en/documents/low-pin-count-interface-specification.pdf">https://www.intel.com/content/dam/www/program/design/us/en/documents/low-pin-count-interface-specification.pdf</a>

## 22.1 Signal Description

Name	Type	Description
GPP_A1 / <b>LAD0</b> / ESPI_IO0	I/O	<b>LPC Multiplexed Command, Address, Data.</b> For LAD0, internal Pull-up is provided.
GPP_A2 / <b>LAD1</b> / ESPI_IO1	I/O	<b>LPC Multiplexed Command, Address, Data.</b> For LAD1, internal Pull-up is provided.
GPP_A3 / <b>LAD2</b> / ESPI_IO2	I/O	<b>LPC Multiplexed Command, Address, Data.</b> For LAD2, internal Pull-up is provided.
GPP_A4 / <b>LAD3</b> / ESPI_IO3	I/O	<b>LPC Multiplexed Command, Address, Data.</b> For LAD3, internal Pull-up is provided.
GPP_A5 / <b>LFRAME#</b> / ESPI_CS#	O	<b>LPC Frame:</b> LFRAME# indicates the start of an LPC cycle, or an abort.
GPP_A9 / <b>CLKOUT_LPC0</b> / ESPI_CLK	O	<b>Low Pin Count (LPC) Clock Outputs:</b> Single-Ended <b>24-MHz</b> output to various single load connectors/ devices.
GPP_A10 / <b>CLKOUT_LPC1</b>	O	<b>Low Pin Count (LPC) Clock Outputs:</b> Single-Ended <b>24-MHz</b> output to various single load connectors/ devices.
GPP_A8 / <b>CLKRUN#</b>	I/O	<b>LPC Clock Run for control of CLKOUT_LPC[1:0]:</b> Connects to peripherals that need to request clock restart or prevention of clock stopping.
GPP_A6 / <b>SERIRQ</b> / ESPI_CS1#	I/O	This signal implements the serial interrupt protocol. <i>Note:</i> An external Pull-up to V3.3S power rail is required.
GPP_A14 / <b>SUS_STAT#</b> / ESPI_RESET#	O	<b>LPC Mode - Suspend Status:</b> This signal is asserted by the PCH to indicate that the system will be entering a low power state soon. This can be monitored by devices with memory that need to switch from normal refresh to suspend refresh mode. It can also be used by other peripherals as an indication that they should isolate their outputs that may be going to powered-off planes.

## 22.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value
<b>LAD[3:0]</b>	Pull-up	15 - 40 kOhm

## 22.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>LAD[3:0]</b>	Primary	Internal Pull-up	Internal Pull-up	Undriven	OFF
<b>LFRAME#</b>	Primary	Driven High	Driven High	Driven Low	OFF
<b>CLKOUT_LPC0</b>	Primary	Toggling	Toggling	Driven Low	OFF
<b>CLKOUT_LPC1</b>	Primary	Toggling	Toggling	Driven Low	OFF
<b>CLKRUN#</b>	Primary	Undriven	Undriven	Undriven	OFF
<i>continued...</i>					

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>SERIRQ</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>SUS_STAT#</b>	Primary	Driven Low	Driven Low	Driven Low	OFF
<i>Note:</i> 1. Reset reference for primary well pins is RSMRST#					

## 22.4 Functional Description

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

### 22.4.1 LPC Cycle Types

The PCH implements the cycle types shown in the table below.

**Table 74. LPC Cycle Types Supported**

Cycle Type	Comment
Memory Read	1 byte only—(Refer the note below)
Memory Write	1 byte only—(Refer the note below)
I/O Read	1 byte only—The PCH breaks up 16-bit and 32-bit processor cycles into multiple 8-bit transfers.
I/O Write	1 byte only—The PCH breaks up 16-bit and 32-bit processor cycles into multiple 8-bit transfers.
Firmware Memory Read	1 byte only
Firmware Memory Write	1 byte only
<i>Note:</i> The PCH provides a single generic memory range (LGMR) for decoding memory cycles and forwarding them as LPC Memory cycles on the LPC bus. The LGMR memory decode range is 64 KB in size, and can be defined as being anywhere in the 4 GB memory space. This range needs to be configured by BIOS during POST to provide the necessary memory resources. BIOS should advertise the LPC Generic Memory Range as Reserved to the OS in order to avoid resource conflict. For larger transfers, the PCH performs multiple 8-bit transfers. If the cycle is not claimed by any peripheral, it is subsequently aborted, and the PCH returns a value of all 1s to the processor. This is done to maintain compatibility with ISA memory cycles where pull-up resistors would keep the bus high if no device responds.	

### 22.4.2 Start Field Definition

Bits[3:0] Encoding	Definition
0000	Start of cycle for a generic target
1111	Stop/Abort: End of a cycle for a target
<i>Note:</i> All other encodings are RESERVED.	

### 22.4.3 Cycle Type/Direction (CYCTYPE + DIR)

The PCH always drives bit0 of this field to 0. The Table below shows the valid bit encodings.

**Table 75. Cycle Type Bit Definitions**

Bits[3:2]	Bit1	Definition
00	0	I/O Read
00	1	I/O Write
01	0	Memory Read
01	1	Memory Read
11	x	Reserved. If a peripheral performing a bus master cycle generates this value, the PCH aborts the cycle.
<i>Note:</i> All other encodings are RESERVED.		

## 22.4.4 Size

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

**Table 76. Transfer Size Bit Definition**

Bits[1:0]	Size
00	8-bit transfer (1 byte)
01	16-bit transfer (2 bytes)
10	Reserved
11	32-bit transfer (4 bytes)

## SYNC

Valid values for the SYNC field are shown in the table below.

## 22.4.5 SYNC Timeout

**Table 77. SYNC Bit Definition**

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

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

### 22.4.6 SYNC Error Indication

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

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

### 22.4.7 LFRAME# Usage

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

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

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

### 22.4.8 I/O Cycles

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

---

#### NOTE

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

---

### 22.4.9 LPC Power Management

#### LPCPD# Protocol

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

---

#### NOTE

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

---

## 22.4.10 Configuration and PCH Implications

### LPC I/F Decoders

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

---

#### NOTE

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

---

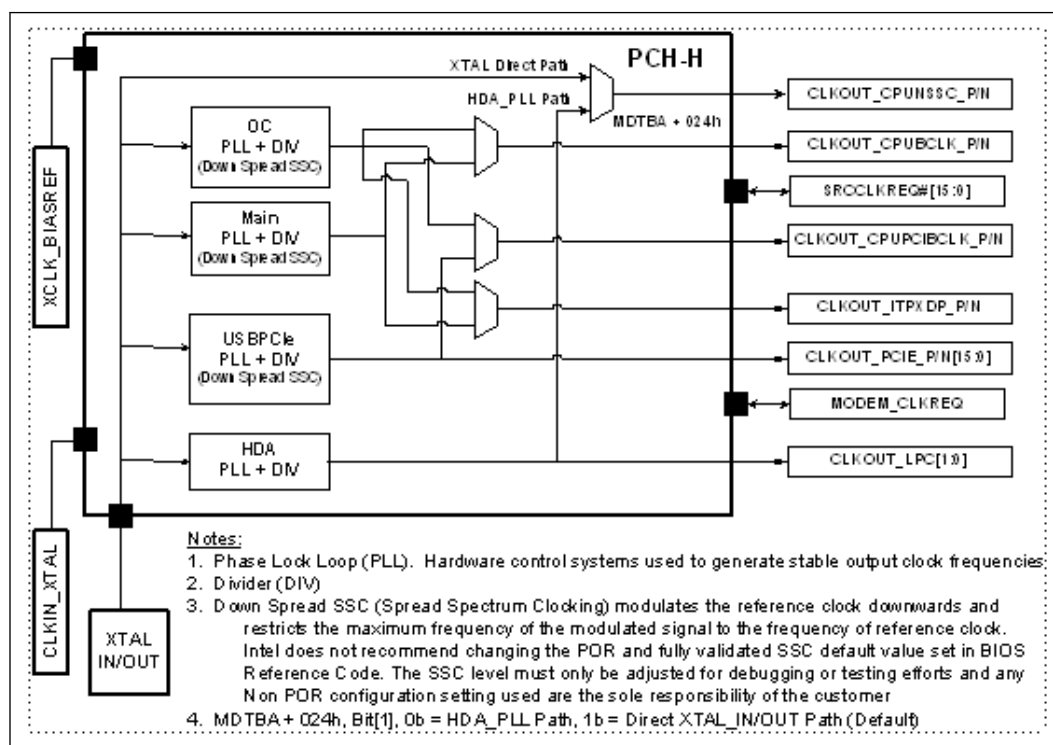
### Turn Around Cycle Behavior

The turn around cycle is defined by the LPC specification as a 2 clock period where the PCH transfers ownership of the LPC LAD[3:0] bus to another device that would like to transfer data. During this turn around cycle it is expected that the state of the bus is to be at high-Z or logic high. During the first clock period of the turnaround cycle the PCH LPC controller will drive an active '1111' on the LAD[3:0] while its output buffers are enabled. At end of the first clock period and start of the second clock period the PCH LPC controller will internally drive LAD[3:0] to '0000' while simultaneously disabling its output buffers to allow the internal pull-up resistors to keep the LAD[3:0] at high-Z. The high-Z condition allows a secondary device to take ownership of the LPC bus after the second clock of the turnaround cycle and drive LAD[3:0] signals. Due to variances in the PCH silicon there may be a delay between the PCH LPC controller disabling its output buffers while internally driving LAD[3:0] to '0000' during the first to second clock transition of the turn-around cycle resulting in a small observable voltage droop on LAD[3:0]. This droop is not expected to cause any bus contention as no devices should be transmitting or reading information on LAD[3:0] during either of the 2 clock periods of the turn around cycle as defined by the LPC specification.

## 23.0 PCH and System Clocks

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

**Figure 32. Integrated Clock Controller (ICC) Diagram**



### 23.1 PCH ICC Clocking Profiles

The PCH ICC hardware includes the following clocking profiles:

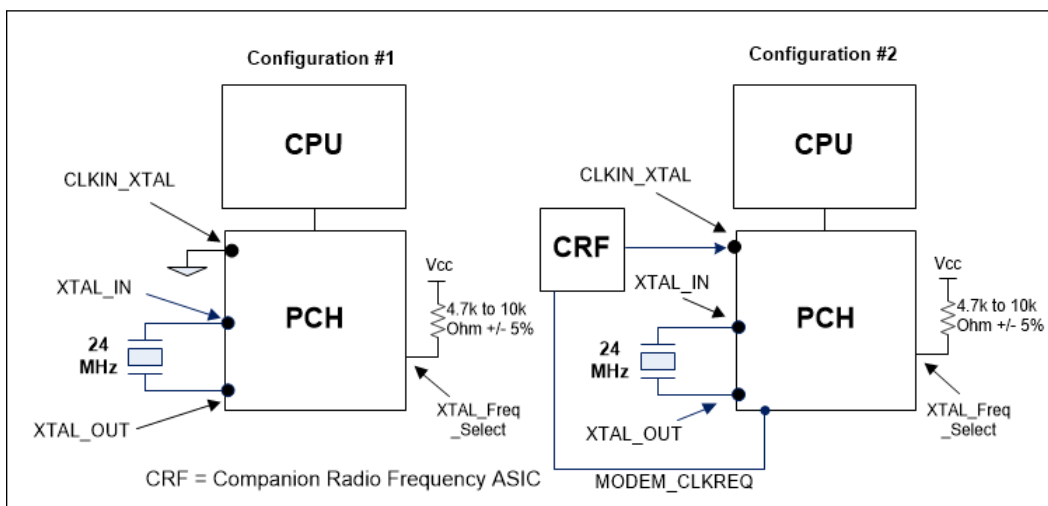
- “Standard” Profile (Figure 32 on page 169)
  - OC PLL = Disabled
  - Main PLL = Enabled with Down Spread Spectrum Clocking (SSC)
  - USBPCIe PLL = Enabled with Down Spread Spectrum Clocking (SSC)
  - HDA PLL = Enabled
- “Adaptive” Profile (Figure 32 on page 169)

- OC PLL = Enabled with Down Spread Spectrum Clocking (SSC) and Under Clocking Capability
- Main PLL = Disabled
- USBPCIe PLL = Enabled with Down Spread Spectrum Clocking (SSC)
- HDA PLL = Enabled
- “Over Clocking” Profile (Figure 32 on page 169)
  - OC PLL = Enabled with Down Spread Spectrum Clocking (SSC) and Over Clocking Capability
  - Main PLL = Disabled
  - USBPCIe PLL = Enabled with Down Spread Spectrum Clocking (SSC)
  - HDA PLL = Enabled

#### NOTE

The Standard ICC Profile is set by default and is the recommended ICC Clocking Profile.

## 23.2 PCH ICC XTAL Input Configurations



## 23.3 Signal Descriptions

Name	Type	SSC Capable	Description
CLKOUT_CPUNSSC_P CLKOUT_CPUNSSC_N	O	No	<b>Filtered Crystal Reference Clock to CPU:</b> 24 MHz differential, filtered crystal reference clock to the processor
CLKOUT_CPUBCLK_P CLKOUT_CPUBCLK_N	O	Yes	<b>Differential Clock to CPU:</b> Reference clock to the processor 100 MHz differential core
CLKOUT_CUPCIBCLK_P CLKOUT_CUPCIBCLK_N	O	Yes	<b>Differential PCIe* Reference Clock to CPU:</b> 100 MHz PCIe* 3.0 specification compliant differential PCIe* reference
continued...			

Name	Type	SSC Capable	Description
			clock to the processor
CLKOUT_ITPXD_P CLKOUT_ITPXD_N	O	Yes	<b>Differential ITP Debug Clock:</b> 100 MHz differential output to processor XDP/ITP connector on the platform
CLKOUT_PCIE_P[15:0] CLKOUT_PCIE_N[15:0]	O	Yes	<b>PCI Express* Clock Output:</b> Serial Reference 100 MHz PCIe* 3.0 specification compliant differential output clocks to PCIe* devices
CLKOUT_LPC[1:0]	O	No	<b>Low Pin Count (LPC) Clock Outputs:</b> Single-Ended 24 MHz output to various single load connectors/devices
SRCCLKREQ#[15:0]	I/O		<b>Clock Request:</b> Serial Reference Clock request signals for PCIe* 100 MHz differential clocks
MODEM_CLKREQ	O		<b>Aggregated XTAL CLKREQ:</b> Used for Companion Radio Frequency ASIC (CRF) device initialization flow indication
XTAL_IN	I		<b>Crystal Input:</b> Input connection for 24 MHz crystal to PCH
XTAL_OUT	O		<b>Crystal Output:</b> Output connection for 24 MHz crystal to PCH
XCLK_BIASREF	I/O		<b>Differential Clock Bias Reference:</b> Used to set BIAS reference for differential clocks
CLKIN_XTAL	I		<b>XTAL Clock Input:</b> Single ended integrated CNV (Connectivity) XTAL clock input
XTAL_Freq_Select (GPP_J4)	I		<b>XTAL Frequency Select:</b> GPP_J4 Pin Strap for XTAL frequency selection. An external pull-up to VCC (1.8V or 3.3V) is required on this strap for 24 MHz XTAL operation
<p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>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.</li> <li>2. The SRCCLKREQ#[15:0] signals can be configured to map to any of the PCH-H PCI Express* Root Ports.</li> <li>3. SRCCLKREQ#[15:0] to CLKOUT_PCIE_P/N[15:0] Mapping Requirements: <ul style="list-style-type: none"> <li>• SRCCLKREQ#[7:0] signals can be mapped to any of the CLKOUT_PCIE_P/N[7:0] differential clock pairs</li> <li>• SRCCLKREQ#[15:8] signals can be mapped to any of the CLKOUT_PCIE_P/N[15:8] differential clock pairs</li> </ul> </li> </ol>			

## 23.4 I/O Signal Pin States

Signal Name	S3/S4/S5	S0 Entry	S0	Deep Sx
CLKOUT_CPUNSSC_P CLKOUT_CPUNSSC_N	OFF (Gated Low)	Bringing up the Clock	Toggling	OFF (Gated Low)
CLKOUT_CPUBCLK_P CLKOUT_CPUBCLK_N	OFF (Gated Low)	Bringing up the Clock	Toggling	OFF (Gated Low)
CLKOUT_CPUPCIBCLK_P CLKOUT_CPUPCIBCLK_N	OFF (Gated Low)	Bringing up the Clock	Toggling	OFF (Gated Low)
CLKOUT_ITPXD_P CLKOUT_ITPXD_N	OFF (Gated Low)	Bringing up the Clock	Toggling	OFF (Gated Low)
CLKOUT_PCIE_P[15:0] CLKOUT_PCIE_N[15:0]	OFF (Gated Low)	Bringing up the Clock	Toggling	OFF (Gated Low)
CLKOUT_LPC[1:0]	OFF (Gated Low)	Bringing up the Clock	Toggling	OFF (Gated Low)
SRCCLKREQ#[15:0]	Un-driven	Un-driven	Driven	OFF
MODEM_CLKREQ	Un-driven	Un-driven	Driven	OFF

## 23.5 General Features

- The PCH Integrated Clock Controller (ICC) generates and supplies all the PCH reference clocks for internal needs and it provides the complete platform system clocking solution.
- All of the ICC PCH internal reference clocks and all of the single-ended and differential clock outputs are generated from an external 24 MHz crystal through the PCH XTAL\_IN/OUT pins, where the crystal accuracy is required to be less than 100 ppm.

---

### NOTE

ppm stands for parts per million, and it indicates how much a crystal's frequency may deviate from the nominal value.

---

- CLKOUT\_PCIE\_P/CLKOUT\_PCIE\_N 100 MHz PCIe\* 3.0 compliant differential output clocks support CLKREQ# based power management
- CLKOUT\_LPC[1:0] single-ended output clocks support CLKRUN# based power management, they require no external loop back clock for internal logic, and they only support a single load configurations.
- System Power Management support includes shutdown of all PCH ICC Phase Locked Loops (PLL), PCH ICC internal and external clocks, and includes the shutdown of the external crystal source.

## 24.0 PCI Express\* (PCIe\*)

- PCH-H supports up to 16 PCIe\* Ports (or devices) and 24 PCIe Lanes, with transfer rates up to 8 GT/s (Gen3)
- Intel® Rapid Storage Technology (Intel® RST) for PCIe Storage Devices
  - Remapped PCIe NVMe\* SSD Non-Windows\* Boot Operating System configurations are not supported
- Interrupt Generation
- PCI Express Power Management
- Latency Tolerance Reporting (LTR)
- Dynamic Link Throttling
- Port 8xh Decode
- PCI Express Separate Reference Clock with Independent Spread Spectrum Clocking (SRIS)
- Advanced Error Reporting
- Single Root I/O Virtualization (SR-IOV) Capability with Access Control Services (ACS) and Alternative Routing ID (ARI)
- SERR# Generation
- PCI Express ExpressCard 1.0 module based hot-plug
- PCI Express TX and RX Lane Polarity Inversion
- End-to-End PCI Express Controller Lane Reversal
- Dynamic Link Width Negotiation as a Target
- Dynamic Speed Change
- 256B Maximum Data Payload Size
- PCIe Subtractive Decode is not supported
  - PCI can still be supported via a PCIe-to-PCI bridge. However, legacy PCI devices (such as PCMCIA or non-plug-and-play device) that need subtractive decode are not supported.
- Common RefClk RX Architecture support
- Precision Time Measurement (PTM)

### 24.1 Signal Description

PCH	Name	Type	Description
PCH-H	PCIE[24:1]_TXP PCIE [24:1]_TXN	0	PCI Express Differential Transmit Pairs
continued...			

PCH	Name	Type	Description
			These are PCI Express* based outbound high-speed differential signals
	PCIE[24:1]_RXP PCIE[24:1]_RXN	I	<b>PCI Express Differential Receive Pairs</b> These are PCI Express* based inbound high-speed differential signals
	PCIE_RCOMP PCIE_RCOMP_N	I	<b>Impedance Compensation Inputs</b>

## 24.2 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately After Reset <sup>1</sup>	S3/S4/S5	Deep Sx
ESPI_IO [3:0]	Primary	Internal Pull- up	Internal Pull-down	Internal Pull-down	OFF
ESPI_CLK	Primary	Internal Pull-down	Driven Low	Driven Low	OFF
ESPI_CS [1:0]#	Primary	Internal Pull- up	Driven High	Driven High	OFF
ESPI_ALERT#	Primary	Internal Pull- up	Driven High	Driven High	OFF
ESPI_RESET#	Primary	Driven Low	Driven High	Driven High	OFF

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

## 24.3 PCI Express\* Port Support Feature Details

Table 78. PCI Express\* Port Feature Details

PCH	Maximum Device (Ports)	Maximum Lanes	PCIe Gen Type	Encoding	Transfer Rate (MT/s)	Theoretical Maximum Bandwidth (GB/s)		
						x1	x2	x4
PCH-H	16	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

Notes: 1. Theoretical Maximum Bandwidth (GB/s) = ((Transfer Rate \* Encoding \* # PCIe Lanes) / 8) / 1000

• Gen3 Example: = ((8000 \* 128/130 \* 4) / 8) / 1000 = 3.94 GB/s

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

Figure 33. Supported PCI Express\* Link Configurations

PCH-H Details	PCIe* Controller #1				PCIe* Controller #2				PCIe* Controller #3 Cycle Router #1				PCIe* Controller #4				PCIe* Controller #5 Cycle Router #3				PCIe* Controller #6 Cycle Router #2			
Flex I/O Lane #	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
PCIe* Lane #	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1x4	RP 1				RP 5				RP 9				RP 13				RP 17				RP 21			
1x4 LR	RP 1				RP 5				RP 9				RP 13				RP 17				RP 21			
2x2	RP 1		RP 3		RP 5		RP 7		RP 9		RP 11		RP 13		RP 15		RP 17		RP 19		RP 21		RP 23	
1x2+2x1	RP 1	RP 3	RP 4		RP 5	RP 7	RP 8		RP 9	RP 11	RP 12		RP 13	RP 15	RP 16		RP 17	RP 19	RP 20		RP 21	RP 23	RP 24	
2x1+1x2	RP 4	RP 3	RP 1		RP 8	RP 7	RP 5		RP 12	RP 11	RP 9		RP 16	RP 15	RP 13		RP 20	RP 19	RP 17		RP 24	RP 23	RP 21	
4x1	RP 1	RP 2	RP 3	RP 4	RP 5	RP 6	RP 7	RP 8	RP 9	RP 10	RP 11	RP 12	RP 13	RP 14	RP 15	RP 16	RP 17	RP 18	RP 19	RP 20	RP 21	RP 22	RP 23	RP 24

1. The PCH PCIe Link Configuration support will vary depending on the PCH SKU.

Refer to the [PCH SKUs](#) on page 16.

2. RP# refers to a specific PCH PCI Express Root Port #; for example RP3 = PCH PCI Express Root Port 3
3. A PCIe Lane is composed of a single pair of Transmit (TX) and Receive (RX) differential pairs, for a total of four data wires per PCIe Lane (such as, PCIE[3]\_TXP/ PCIE[3]\_TXN and PCIE[3]\_RXP/ PCIE[3]\_RXN make up PCIe Lane 3). A connection between two PCIe devices is known as a PCIe Link, and is built up from a collection of one or more PCIe Lanes which make up the width of the link (such as bundling two 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).
4. The PCIe Lanes can be configured independently from one another but the maximum number of configured Root Ports (Devices) must not be exceeded.
  - PCH-H: A maximum of 16 PCIe Root Ports (or devices) can be enabled
  - When a GbE Port is enabled, the maximum number of PCIe Ports (or devices) that can be enabled reduces based off the following:
    - PCH-H: Max PCIe Ports (or devices) = 16 - GbE (0 or 1)
5. Unidentified lanes within a PCIe Link Configuration are disabled but their physical lanes are used for the identified Root Port
6. Supports Remapped (Intel® Rapid Storage Technology) PCIe Storage Devices
  - Specific PCIe lanes can be used for a x2 or x4 Intel® Rapid Storage Technology Remapped PCIe NVMe SSD or a x2 or x4 Next Generation Intel® Optane™ Memory Device.
  - Not all PCH PCIe lanes support Intel® RST for PCIe Storage Devices. Refer to [PCH SKUs](#) on page 16 for specific PCH SKU PCIe lane support
7. Lane Reversal Supported Motherboard PCIe Configurations = 1x4 and 2x1+1x2
  - The 2x1+1x2 configuration is enabled by setting the PCIe Controller soft straps to 1x2+2x1 with Lane Reversal Enabled.
  - 1x4 = 1x4 with Lane Reversal Disabled, 1x4 LR = 1x4 with Lane Reversal Enabled.
  - 2x2 = 2x2 with Lane Reversal Disabled, 2x2 LR = 2x2 with Lane Reversal Enabled.
8. For unused SATA/PCIe Combo Lanes, Flex I/O Lanes that can be configured as PCIe or SATA, the lanes must be statically assigned to SATA or PCIe via the SATA/PCIe Combo Port Soft Straps. These unused SATA/PCIe\* Combo Lanes must not be assigned as polarity based.
  - Refer to [Flexible I/O](#) on page 26 chapter for SATA/PCIe Combo Lane identification.

### 24.3.1 Intel® Rapid Storage Technology (Intel® RST) for PCIe Storage

Intel® Rapid Storage Technology for PCIe Storage includes the PCH PCIe Controller Remapping Hardware, also referred to as Cycle Routers, and the Intel® RST Driver. The Remapping Hardware is a PCH PCIe Controller architecture feature that works with the Intel® RST Driver to control and remap PCIe storage devices to the PCH AHCI SATA Controller.

The PCH has multiple PCIe Controllers where some, not all, of these Controllers have the Remapping Hardware. These specific PCIe Controllers along with the Intel® RST Driver handle the remapping for x2 or x4 PCIe storage devices. Special care must be taken to make sure the correct PCH PCIe Lanes are used that are associated with these specific PCIe Controllers. The Intel® RST for PCIe Storage controller, configuration, and lane support will vary depending on the PCH SKU.

## 24.3.2 Interrupt Generation

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

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

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

**Table 79. MSI Versus PCI IRQ Actions**

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

## 24.3.3 PCI Express\* Power Management

### S3/S4/S5 Support

Software initiates the transition to S3/S4/S5 by performing an I/O write to the Power Management Control register in the SoC. After the I/O write completion has been returned to the processor, the Power Management Controller will signal each root port to send a PME\_Turn\_Off message on the downstream link. The device attached to the link will eventually respond with a PME\_TO\_Ack followed by sending a PM\_Enter\_L23 DLLP (Data Link Layer Packet) request to enter L23. The Express ports and Power Management Controller take no action upon receiving a PME\_TO\_Ack. When all the Express port links are in state L23, the Power Management Controller will proceed with the entry into S3/S4/S5.

Prior to entering S3, software is required to put each device into D3<sub>HOT</sub>. When a device is put into D3<sub>HOT</sub>, it will initiate entry into a L1 link state by sending a PM\_Enter\_L1 DLLP. Under normal operating conditions when the root ports sends the PME\_Turn\_Off message, the link will be in state L1. However, when the root port is instructed to send the PME\_Turn\_Off message, it will send it whether or not the link was in L1. Endpoints attached to the PCH can make no assumptions about the state of the link prior to receiving a PME\_Turn\_Off message.

### Device Initiated PM\_PME Message

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

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

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

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

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

### SMI/SCI Generation

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

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

### Latency Tolerance Reporting (LTR)

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

---

#### NOTE

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

---

### 24.3.4 Dynamic Link Throttling

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

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

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

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

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

The period of the Throttling Window is configurable through the TNPT.TP field. The Throttling Window is always referenced from the time a new Throttle State change indication is received by the Root Port or from the time the throttling is enabled by the configuration register. The Throttling Window and Throttling Zone timers continue to behave the same as in L0 or L0s even if the link transitions to other LTSSM states, except for L1, L23\_Rdy and link down. For L1 case, the timer is allowed to be stopped and hardware is allowed to re-start the Throttling Window and the corresponding Throttling Zone timers on exit from L1.

### 24.3.5 Port 8xh Decode

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

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

### 24.3.6 Separate Reference Clock with Independent SSC (SRIS)

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

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

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

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

### 24.3.7 Advanced Error Reporting

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

### 24.3.8 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.

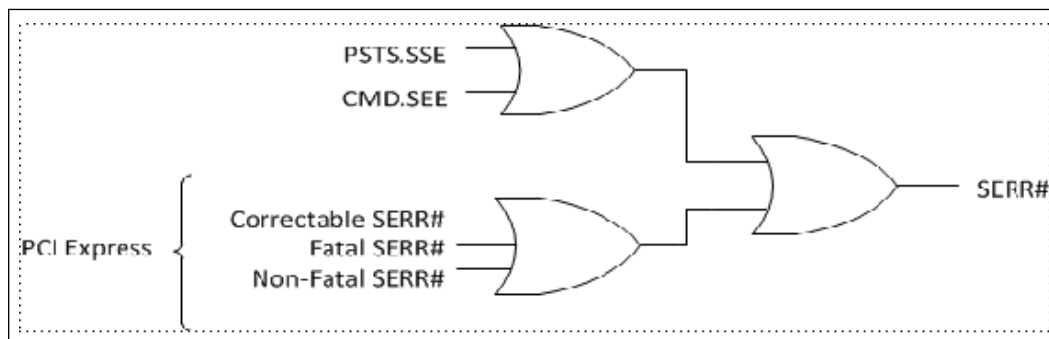
#### Access Control Services (ACS)

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.

### 24.3.9 SERR# Generation

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

**Figure 34. Generation of SERR# to Platform**



### 24.3.10 Hot-Plug

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

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

#### Presence Detection

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

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

#### SMI/SCI Generation

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

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

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

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

### 24.3.11 PCI Express\* Lane Polarity Inversion

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

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

### 24.3.12 PCI Express\* Controller Lane Reversal

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

---

#### NOTES

- Lane Reversal Supported Motherboard PCIe Configurations = 1x4, 2x1+1x2, and 2x2
    - The 2x1+1x2 configuration is enabled by setting the PCIe Controller soft straps to 1x2+2x1 with Lane Reversal Enabled
  - PCI Express Controller Lane Reversal is not the same as PCI Express Lane Polarity Inversion.
- 

### 24.3.13 Precision Time Measurement (PTM)

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

All of the PCH PCIe Controllers and their assigned Root Ports support PTM where each Root Port can have PTM enabled or disabled individually from one another. PCIe Root Ports transmit the lower byte [7:0] of the Propagation Delay Field first instead of the upper byte [31:24] within their PTM DelayResponseD (Response with Data) messages.

## 25.0 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\_S3# and PLTRST#); configure, manage and respond to wake events; aggregate and report latency tolerance information for devices and peripherals connected to and integrated into the PCH.

### NOTE

DeepS3 is not supported in this PCH. In this document DeepSx refers to DeepS4 and DeepS5 states.

### Acronyms

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

### References

Specification	Location
Advanced Configuration and Power Interface (ACPI)	<a href="http://www.acpi.info/spec.htm">http://www.acpi.info/spec.htm</a>

## 25.1 Signal Description

Name	Type	Description
<b>ACPRESENT</b> /GPD1	I	<b>ACPRESENT:</b> This input pin indicates when the platform is plugged into AC power or not. In addition to the previous Intel® CSME to EC communication, the PCH uses this information to implement the Deep Sx policies. For example, the platform may be configured to enter Deep Sx when in S4 or S5 and only when running on battery. This is powered by Deep Sx Well.
<b>BATLOW#</b> /GPD0	I	<b>Battery Low:</b> An input from the battery to indicate that there is insufficient power to boot the system. Assertion will prevent wake from S4/S5 states and exit from Deep Sx state. This signal can also be enabled to cause an SMI# when asserted. This signal is multiplexed with GPD0. For any platform not using this pin functionality, this signal must be tied high to VCCDSW_3P3. 1. An external pull-up resistor to VCCDSW_3P3 is required.
<i>continued...</i>		

Name	Type	Description
<b>CPU_C10_GATE#</b> / GPP_J1	O	External Power Gate control for VCCIO, VCCSTG and VCCPLL_OC during C10. When asserted, VCCIO and VCCSTG can be 0V, however the power good indicators for these rails must remain asserted.
<b>DRAM_RESET#</b>	OD	<b>System Memory DRAM Reset:</b> Active low reset signal to DRAM. <i>Note:</i> An external pull-up to the DRAM power plane is required.
<b>DSW_PWROK</b>	I	<b>DSW PWROK:</b> Power OK Indication for the VCCDSW_3P3 voltage rail. This input is tied together with RSMRST# on platforms that do not support Deep Sx. <i>Note:</i> This signal is in the RTC well.
<b>LAN_WAKE#</b> /GPD2	I	<b>LAN WAKE:</b> is an active low wake indicator from the GbE PHY. <i>Note:</i> External pull-up required.
<b>LANPHYPC</b> /GPD11	O	<b>LAN PHY Power Control:</b> LANPHYPC is used to indicate that power needs to be restored to the Platform LAN Connect Device.
<b>PCH_PWROK</b>	I	<b>PCH Power OK:</b> When asserted, PCH_PWROK is an indication to the PCH that all of its core power rails have been stable for at least 5 ms. PCH_PWROK can be driven asynchronously. When PCH_PWROK is negated, the PCH asserts PLTRST#. <i>Note:</i> PCH_PWROK must not glitch, even if RSMRST# is low.
<b>PLTRST#</b> /GPP_B13	O	<b>Platform Reset:</b> The PCH asserts PLTRST# to reset devices on the platform (such as SIO, LAN, processor, and so forth.). The PCH asserts PLTRST# low in Sx states and when a cold, warm, or global reset occurs. The PCH de-asserts PLTRST# upon exit from Sx states and the aforementioned resets. There is no guaranteed minimum assertion time for PLTRST#. <i>Note:</i> PCIe* specification requires that the power rails associated with PCIe* (typically the 3.3 V, 5 V, and 12 V core well rails) have been valid for 100 ms prior to PLTRST# de-assertion. System designers must ensure the requirement is met on the platform.
<b>PME#</b> /GPP_A11/ SD_VDD2_PWR_EN#	I/OD	<b>Power Management Event:</b> Driven by devices to wake the system or issue SCI.
<b>PWRBTN#</b> /GPD3	I	<b>Power Button:</b> The Power Button will cause SMI# or SCI to indicate a system request to go to a sleep state. If the system is already in a sleep state, this signal will cause a wake event. If PWRBTN# is pressed for more than 4 seconds (default; timing is configurable), this will cause an unconditional transition (power button override) to the S5 state. Override will occur even if the system is in the S3-S4 states. This signal has an internal Pull-up resistor and has an internal 16 ms de-bounce on the input. <i>Note:</i> 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.
<b>RSMRST#</b>	I	<b>Primary Well:</b> This signal is used for resetting the resume power plane logic. This signal must be asserted for at least 10ms after the suspend power wells are valid. When de-asserted, this signal is an indication that the primary power wells are stable.
<b>SLP_A#</b> /GPD6	O	<b>SLP_A#:</b> Signal asserted when the Intel® CSME platform goes to M-Off or M3-PG. Depending on the platform, this pin may be used to control power to various devices that are part of the Intel® CSME sub-system in the platform. If you are not using SLP_A# for any functional purposes on your platform, or can tolerate lack of minimum assertion time, program the "SLP_A# minimum assertion width" value to the minimum.
<b>SLP_LAN#</b>	O	<b>LAN Sub-System Sleep Control:</b> When SLP_LAN# is de-asserted it indicates that the PHY device must be powered. When SLP_LAN# is asserted, power can be shut off to the PHY device. SLP_LAN# will always be de-asserted in S0 and anytime SLP_A# is de-asserted.
<i>continued...</i>		

Name	Type	Description
<b>SLP_WLAN#</b> / GPD9	O	<b>WLAN Sub-System Sleep Control:</b> When SLP_WLAN# is asserted, power can be shut off to the external wireless LAN device. SLP_WLAN# will always be de-asserted in S0. SLP_WLAN# shall not be used to shut down power to integrated connectivity (CNVi). 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.
<b>SLP_S0#</b> /GPP_B12	O	<b>S0 Sleep Control:</b> When PCH is idle and processor is in C10 state, this pin will assert to indicate VR controller can go into a light load mode. This signal can also be connected to EC for other power management related optimizations.
<b>SLP_S3#</b> /GPD4	O	<b>S3 Sleep Control:</b> SLP_S3# is for power plane control. This signal shuts off power to all non-critical systems when in S3 (Suspend To RAM), S4 (Suspend to Disk), or S5 (Soft Off) states.
<b>SLP_S4#</b> /GPD5	O	<b>S4 Sleep Control:</b> SLP_S4# is for power plane control. This signal shuts power to all non-critical systems when in the S4 (Suspend to Disk) or S5 (Soft Off) state. <i>Note:</i> This pin must be used to control the DRAM power in order to use the PCH DRAM power-cycling feature.
<b>SLP_S5#</b> /GPD10	O	<b>S5 Sleep Control:</b> SLP_S5# is for power plane control. This signal is used to shut power off to all non-critical systems when in the S5 (Soft Off) states.
<b>SLP_SUS#</b>	O	<b>Deep Sx Indication:</b> When asserted (driven low), this signal indicates PCH is in Deep Sx state where 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. If Deep Sx is not supported, then this pin can be left unconnected. <i>Note:</i> This pin is in the DSW power well.
<b>SUSACK#</b> /GPP_A15	I	<b>SUSACK#:</b> If Deep Sx is supported, the EC/motherboard controlling logic must change SUSACK# to match SUSWARN# once the EC/motherboard controlling logic has completed the preparations discussed in the description for the SUSWARN# pin. <i>Note:</i> SUSACK# is only required to change in response to SUSWARN# if Deep Sx is supported by the platform.
<b>SUSCLK</b> /GPD8	O	<b>Suspend Clock:</b> This clock is a digitally buffer version of the RTC clock.
<b>SUSWARN#</b> / SUSPWRDNACK/ GPP_A13	O	<b>SUSWARN#:</b> This pin asserts low when the PCH is planning to enter the Deep Sx power state and remove Primary power (using SLP_SUS#). The EC/motherboard controlling logic must observe edges on this pin, preparing for primary well power loss on a falling edge and preparing for Primary well related activity (host/Intel® CSME wakes and runtime events) on a rising edge. SUSACK# must be driven to match SUSWARN# once the above preparation is complete. SUSACK# should be asserted within a minimal amount of time from SUSWARN# assertion as no wake events are supported if SUSWARN# is asserted but SUSACK# is not asserted. Platforms supporting Deep Sx, but not wishing to participate in the handshake during wake and Deep Sx entry may tie SUSACK# to SUSWARN#. This pin is multiplexed with SUSPWRDNACK since it is not needed in Deep Sx supported platforms.
<b>SUSPWRDNACK</b> / SUSWARN#/GPP_A13	O	<b>SUSPWRDNACK:</b> 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.
<b>SX_EXIT_HOLDOFF#</b> / GPP_A12 /ISH_GP6 / BM_BUSY#	I	<b>Sx Exit Holdoff Delay:</b> Delay exit from Sx state after SLP_A# is de-asserted. Refer to <a href="#">Sx Exit Holdoff#</a> on page 202 for more details. When eSPI is enabled, SX_EXIT_HOLDOFF# functionality is not available, and assertion of the signal will not impact Sx exit flows.
<b>continued...</b>		

Name	Type	Description
<b>SYS_PWROK</b>	I	<b>System Power OK:</b> This generic power good input to the PCH is driven and utilized in a platform-specific manner. While PCH_PWROK always indicates that the primary wells of the PCH are stable, SYS_PWROK is used to inform the PCH that power is stable to some other system component(s) and the system is ready to start the exit from reset.
<b>SYS_RESET#</b>	I	<b>System Reset:</b> This pin forces an internal reset after being de-bounced.
<b>VRALERT#</b> /GPP_B2	I	<b>VR Alert:</b> ICC Max. throttling indicator from the PCH voltage regulators. VRAlert# pin allows the VR to force throttling to prevent an over current shutdown.
<b>WAKE#</b>	I/OD	<b>PCI Express* Wake Event in Sx:</b> Input Pin in Sx. Sideband wake signal on PCI Express* asserted by components requesting wake up. <i>Note:</i> This is Output pin during S0ix states hence this pin can not be used to wake up the system during S0ix states. <i>Note:</i> External pull-up required.
<b>CLKRUN#</b> /GPP_A8	I/OD	<b>LPC Clock Run:</b> Used to control CLKOUT_LPC[1:0]. Connects to peripherals that need to request clock restart or prevention of clock stopping.
<b>SUS_STAT#</b> / ESPI_RESET#/GPP_A14	O	<b>LPC Mode - Suspend Status:</b> This signal is asserted by the PCH to indicate that the system will be entering a low power state soon. This can be monitored by devices with memory that need to switch from normal refresh to suspend refresh mode. It can also be used by other peripherals as an indication that they should isolate their outputs that may be going to powered-off planes. <i>Note:</i> In eSPI Mode, this signal functions as ESPI Reset#. Reset signal from PCH to eSPI slave.

## 25.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value
<b>ACPRESENT</b>	Pull-down	15 kOhm - 40 kOhm
<b>LAN_WAKE#</b>	Pull-down	15 kOhm - 40 kOhm
<b>PWRBTN#</b>	Pull-up	20 kOhm +/- 30%
<b>SUSACK#</b>	Pull-up	20 kOhm +/- 30%
<b>WAKE#</b>	Pull-down	15 kOhm - 40 kOhm
<i>Note:</i> Pull-down is configurable and can be enabled in Deep Sx state.		

## 25.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>16</sup>	Immediately after Reset <sup>16</sup>	S3/S4/S5	Deep Sx
<b>ACPRESENT<sup>6,10,15</sup></b>	DSW	Undriven /Driven Low <sup>4</sup>	Undriven	Undriven	Undriven/Internal Pull-down <sup>8</sup>
<b>BATLOW#</b>	DSW	Undriven	Undriven	Undriven	OFF
<b>DRAM_RESET#</b>	DSW	Undriven	Undriven	Undriven	Undriven
<b>DSW_PWROK</b>	RTC	Undriven	Undriven	Undriven	Undriven
<b>LANPHYPC<sup>10,16</sup></b>	DSW	Undriven	Undriven	Undriven <sub>7</sub>	Undriven <sub>7</sub>
<b>continued...</b>					

Signal Name	Power Plane	During Reset <sup>16</sup>	Immediately after Reset <sup>16</sup>	S3/S4/S5	Deep Sx
<b>LAN_WAKE#<sup>15</sup></b>	DSW	Undriven	Undriven	Undriven	Undriven/Internal Pull-down <sup>8</sup>
<b>PCH_PWROK</b>	RTC	Undriven	Undriven	Undriven	Undriven
<b>PLTRST#<sup>16</sup></b>	Primary	Driven Low	Driven High	Driven Low	OFF
<b>PME#<sup>15</sup></b>	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up	OFF
<b>PWRBTN#<sup>15</sup></b>	DSW	Internal Pull-up	Internal Pull-up	Internal Pull-up	Internal Pull-up
<b>RSMRST#</b>	RTC	Undriven	Undriven	Undriven	Undriven
<b>SUSACK#<sup>15</sup></b>	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up	OFF
<b>SUSCLK<sup>10,16</sup></b>	DSW	Driven Low	Toggling	Toggling	Toggling <sup>10</sup>
<b>SUSWARN# / SUSPWRDNACK<sup>6,10, 16</sup></b>	Primary	Driven Low	Driven Low	Driven Low <sup>5</sup>	OFF
<b>SLP_A#<sup>6,16</sup></b>	DSW	Driven Low	Driven High	Driven High/ Driven Low <sup>12</sup>	Driven High/ Driven Low <sup>12</sup>
<b>SLP_S0#<sup>15</sup></b>	Primary	Driven High	Driven High	Driven High	OFF
<b>SLP_S3#<sup>6,16</sup></b>	DSW	Driven Low	Driven High	Driven Low	Driven Low
<b>SLP_S4#<sup>6,16</sup></b>	DSW	Driven Low	Driven High	Driven Low	Driven Low <sup>9</sup>
<b>SLP_S5#<sup>6,16</sup></b>	DSW	Driven Low	Driven High	Driven High/ Driven Low <sup>3</sup>	Driven High/ Driven Low <sup>9</sup>
<b>SLP_LAN#<sup>6</sup></b>	DSW	Driven Low	Driven Low	Driven High/ Driven Low <sup>7</sup>	Driven High/ Driven Low <sup>7</sup>
<b>SLP_WLAN#<sup>6,16</sup></b>	DSW	Driven Low	Driven Low	Driven High/ Driven Low <sup>7</sup>	Driven High/ Driven Low <sup>7</sup>
<b>SLP_SUS#<sup>6</sup></b>	DSW	Driven Low	Driven High	Driven High	Driven Low
<b>SX_EXIT_HOLDOFF#<sup>15</sup></b>	Primary	Undriven	Undriven	Undriven	OFF
<b>SYS_PWROK<sup>13</sup></b>	Primary	Undriven	Undriven	Undriven	OFF
<b>SYS_RESET#<sup>13</sup></b>	Primary	Undriven	Undriven	Undriven	OFF
<b>continued...</b>					

Signal Name	Power Plane	During Reset <sup>16</sup>	Immediately after Reset <sup>16</sup>	S3/S4/S5	Deep Sx
<b>VRALERT#<sup>15</sup></b>	Primary	Undriven	Undriven	Undriven	OFF
<b>WAKE#<sup>13</sup></b>	DSW	Undriven	Undriven	Undriven	Undriven/Internal Pull-down

Notes:

1. Driven High during S0 and driven Low during S0 CS.
2. In non-Deep Sx mode, pin is driven low.
3. Based on wake events and Intel® CSME state. SUSWRDNACK is always '0' while in M0 or M3, but can be driven to '0' or '1' while in M0ff state. SUSWRDNACK is the default mode of operation. If Deep Sx is supported, then subsequent boots will default to SUSWRN#.
4. The pin requires glitch-free output sequence. The pad should only be pulled low momentarily when the corresponding buffer power supply is not stable.
5. Based on wake event and Intel® CSME state.
6. Pull-down is configurable and can be enabled in Deep Sx state; refer to DSX\_CFG register for more details.
7. When platform enters Deep Sx, the SLP\_S4# and SLP\_S5# pin will retain the value it held prior to Deep Sx entry.
8. Internal weak pull-down resistor is enabled during power sequencing.
9. N/A
10. Pin state is a function of whether the platform is configured to have Intel® CSME on or off in Sx.
11. Output High-Z, not glitch free.
12. Output High-Z, glitch free with ~120 kOhm Pull-down during respective power sequencing.
13. Output High-Z, not glitch free.
14. Output High-Z, glitch free with ~20 kOhm Pull-down during respective power sequencing.
15. Output High-Z, glitch free with ~20 kOhm Pull-up during respective power sequencing.
16. Reset reference for primary well pins is RSMRST#, DSW well pins is DSW\_PWROK, and RTC well pins is RTCRST#.

## 25.4 Functional Description

Topics Covered:

- Features
- PCH and System Power States
- System Power Planes
- SMI#/SCI Generation
- C-States
- Dynamic 24 MHz Clock Control
- 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

### 25.4.1 Features

- Support for *Advanced Configuration and Power Interface (ACPI)* providing power and thermal management
  - ACPI 24-Bit Timer SCI and SMI# Generation

- PCI PME# signal for Wake Up from Low-Power states
- System Sleep State Control
  - ACPI S3 state – Suspend to RAM (STR)
  - ACPI S4 state – Suspend-to-Disk (STD)
  - ACPI G2/S5 state – Soft Off (SOFF)
  - Power Failure Detection and Recovery
  - Deep Sx
- Intel® Converged Security and Management Engine Power Management Support
  - Wake events from the Intel® Converged Security and Management Engine (enabled from all S-States including Catastrophic S5 conditions)
- SLP\_S0# signal for external platform VR power gating or EC power management handling during lower power condition

---

**NOTE**

Power Management timers accuracy is ~4.2% which will affect energy reporting counters, power button override duration etc.

---

## 25.4.2 PCH and System Power States

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

**Table 80. General Power States for Systems Using the PCH**

State / Substates	Legacy Name/Description
G0/S0/C0	<b>Full On:</b> Processor operating. Individual devices may be shut down or be placed into lower power states to save power.
G0/S0/Cx	<b>Cx State:</b> Cx states are processor power states within the S0 system state that provide for various levels of power savings. The processor manages c-state itself. The actual c-state is not passed to the PCH. Only c-state related messages are sent to the PCH and PCH will base its behavior on the actual data passed.
G1/S3	<b>Suspend-To-RAM (STR):</b> The system context is maintained in system DRAM, but power is shut off to non-critical circuits. Memory is retained and refreshes continue. All external clocks stop except RTC.
G1/S4	<b>Suspend-To-Disk (STD):</b> The context of the system is maintained on the disk. All power is then shut off to the system except for the logic required to resume.
G2/S5	<b>Soft Off (SOFF):</b> System context is not maintained. All power is shut off except for the logic required to restart. A full boot is required when waking.
<i>continued...</i>	

State / Substates	Legacy Name/Description
S0ix	S0 idle states are often referred as S0ix states. CPU PKG C-states and platform latency tolerance will decide when to take the aggressive power management actions.
Deep Sx	<b>Deep Sx:</b> An optional low power state where system context may or may not be maintained depending upon entry condition. All power is shut off except for minimal logic that allows exiting Deep Sx. If Deep Sx state was entered from S4 state, then the resume path will place system back into S4. If Deep Sx state was entered from S5 state, then the resume path will place system back into S5.
G3	<b>Mechanical OFF (M-Off):</b> System context not maintained. All power is shut off except for the RTC. No "Wake" events are possible. This state occurs if the user removes the main system batteries in a mobile system, turns off a mechanical switch, or if the system power supply is at a level that is insufficient to power the "waking" logic. When system power returns, transition will depend on the state just prior to the entry to G3 and the AFTERG3_EN bit in the General Power Management Configuration (GEN_PMCON) register. Refer to <a href="#">Table 86</a> on page 197 for more details.

The following table shows the transitions rules among the various states.

#### NOTE

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

**Table 81. State Transition Rules for the PCH**

Present State	Transition Trigger	Next State
G0/S0/C0	<ul style="list-style-type: none"> <li>DMI Msg</li> <li>SLP_EN bit set</li> <li>Power Button Override<sup>3,5</sup></li> <li>Mechanical Off/Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>G0/S0/Cx</li> <li>G1/Sx or G2/S5 state</li> <li>G2/S5</li> <li>G3</li> </ul>
G0/S0/Cx	<ul style="list-style-type: none"> <li>DMI Msg</li> <li>Power Button Override<sup>3,5</sup></li> <li>Mechanical Off/Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>G0/S0/C0</li> <li>S5</li> <li>G3</li> </ul>
G1/S3	<ul style="list-style-type: none"> <li>Any Enabled Wake Event</li> <li>Power Button Override<sup>3,5</sup></li> <li>Mechanical Off/Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>G0/S0/C0<sup>2</sup></li> <li>G2/S5</li> <li>G3</li> </ul>
G1/S4	<ul style="list-style-type: none"> <li>Any Enabled Wake Event</li> <li>Power Button Override<sup>3,5</sup></li> <li>Conditions met as described in Entry into Deep Sx and Exit from Deep Sx in <a href="#">Deep Sx</a> on page 197</li> <li>Mechanical Off/Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>G0/S0/C0<sup>2</sup></li> <li>G2/S5</li> <li>Deep S4</li> <li>G3</li> </ul>
G2/S5	<ul style="list-style-type: none"> <li>Any Enabled Wake Event</li> <li>Conditions met as described in Entry into Deep Sx and Exit from Deep Sx in <a href="#">Deep Sx</a> on page 197</li> <li>Mechanical Off/Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>G0/S0/C0<sup>2</sup></li> <li>Deep S5</li> <li>G3</li> </ul>
G2/Deep Sx	<ul style="list-style-type: none"> <li>Any Enabled Wake Event</li> <li>ACPRESENT Assertion</li> </ul>	<ul style="list-style-type: none"> <li>G0/S0/C0<sup>2</sup></li> </ul>

*continued...*

Present State	Transition Trigger	Next State
	<ul style="list-style-type: none"> <li>Mechanical Off/Power Failure</li> <li>Power Button Override</li> </ul>	<ul style="list-style-type: none"> <li>G1/S4 or G2/S5 (Refer to Exit from Deep Sx in <a href="#">Deep Sx</a> on page 197)</li> <li>G3</li> <li>G2/S5</li> </ul>
G3	<ul style="list-style-type: none"> <li>Power Returns</li> </ul>	<ul style="list-style-type: none"> <li>S0/C0 (reboot) or G2/S5<sup>4</sup> (stay off until power button pressed or other wake event)<sup>1,2</sup></li> </ul>
<p><i>Notes:</i> 1. Some wake events can be preserved through power failure.</p> <p>2. Transitions from the S3–S5 or G3 states to the S0 state are deferred until BATLOW# is inactive in mobile configurations.</p> <p>3. Includes all other applicable types of events that force the host into and stay in G2/S5.</p> <p>4. If the system was in G1/S4 before G3 entry, then the system will go to S0/C0 or G1/S4.</p> <p>5. Upon entry to S5 due to a power button override, if Deep S5 is enabled and conditions are met per <a href="#">Deep Sx</a> on page 197, the system will transition to Deep S5.</p>		

### 25.4.3 System Power Planes

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

#### NOTE

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

**Table 82. System Power Plane**

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

Plane	Controlled By	Description
Primary/ Suspend Well	SLP_SUS#	This signal is asserted when the Primary/Suspend 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 0V.
DEVICE[n]	Implementation Specific	Individual subsystems may have their own power plane. For example, GPIO signals may be used to control the power to disk drives, audio amplifiers, or the display screen.

## 25.4.4 SMI#/SCI Generation

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

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

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

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

The following table shows which events can cause an SMI and SCI.

### NOTE

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

**Table 83. Causes of SMI and SCI**

Cause	SCI	SMI	Additional Enables <sup>1</sup>	Where Reported
PME#	Yes	Yes	PME_EN=1	PME_STS
PME_B0 (Internal, Bus 0, PME-Capable Agents)	Yes	Yes	PME_B0_EN=1	PME_B0_STS
PCI Express* PME Messages	Yes	Yes	PCI_EXP_EN=1 (Not enabled for SMI)	PCI_EXP_STS
PCI Express Hot-Plug Message	Yes	Yes	HOT_PLUG_EN=1 (Not enabled for SMI)	HOT_PLUG_STS
Power Button Press	Yes	Yes	PWRBTN_EN=1	PWRBTN_STS
Power Button Override (Note 6)	Yes	No	None	PRBTNOR_STS
<i>continued...</i>				

Cause	SCI	SMI	Additional Enables <sup>1</sup>	Where Reported
RTC Alarm	Yes	Yes	RTC_EN=1	RTC_STS
ACPI Timer overflow (2.34 seconds)	Yes	Yes	TMROF_EN=1	TMROF_STS
GPIO (Note 8)	Yes	Yes		
LAN_WAKE#	Yes	Yes	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 (Note 7)	Yes	No	GBL_EN=1	GBL_STS
GBL_RLS written to	No	Yes	BIOS_EN=1	BIOS_STS
Write to B2h register	No	Yes	APMC_EN = 1	APM_STS
Periodic timer expires	No	Yes	PERIODIC_EN=1	PERIODIC_STS
64 ms timer expires	No	Yes	SWSMI_TMR_EN=1	SWSMI_TMR_STS
Enhanced USB Legacy Support Event	No	Yes	LEGACY_USB2_EN = 1	LEGACY_USB2_STS
Serial IRQ SMI reported	No	Yes	None	SERIRQ_SMI_STS
Device monitors match address in its range	No	Yes	Refer to DEVTRAP_STS register description	DEVTRAP_STS
SMBus Host Controller	No	Yes	SMB_SMI_EN, Host Controller Enabled	SMBus host status reg.
SMBus Slave SMI message	No	Yes	None	SMBUS_SMI_STS
SMBus SMBALERT# signal active	No	Yes	None	SMBUS_SMI_STS
SMBus Host Notify message received	No	Yes	HOST_NOTIFY_INTREN	SMBUS_SMI_STS, HOST_NOTIFY_STS
BATLOW# assertion	Yes	Yes	BATLOW_EN=1	BATLOW_STS
Access microcontroller 62h/66h	No	Yes	MCSMI_EN	MCSMI_STS
SLP_EN bit written to 1	No	Yes	SMI_ON_SLP_EN=1	SMI_ON_SLP_EN_STS
SPI Command Completed	No	Yes	None	SPI_SMI_STS
eSPI SCI/SMI Request	Yes	Yes	eSPI_SCI_EN For SMI, refer to <a href="#">enhanced Serial Peripheral Interface (eSPI)</a> on page 117.	eSPI_SCI_STS eSPI_SMI_STS
<b>continued...</b>				

Cause	SCI	SMI	Additional Enables <sup>1</sup>	Where Reported
Software Generated GPE	Yes	Yes	SWGPE_EN=1	SWGPE_STS
Intel® CSME	Yes	Yes	ME_SCI_EN=1 ME_SCI_EN=0; ME_SMI_EN=1;	ME_SCI_STS ME_SMI_STS
GPIO Lockdown Enable bit changes from '1' to '0'	No	Yes	GPIO_UNLOCK_SMI_EN=1	GPIO_UNLOCK_SMI_STS
USB 3.2(xHCI) SMI Event	No	Yes	XHCI_SMI_EN=1	XHCI_SMI_STS
Wake Alarm Device Timer	Yes	Yes	WADT_EN	WADT_STS
ISH	Yes	No	ISH_EN	ISH_STS
RTC update-in-progress	No	Yes	Refer to I/O Trap Register section	RTC_UIP_SMI_STS
SIO SMI events	No	Yes	SIP_SMI_EN	SIO_SMI_STS
SCC	No	Yes	SCC_SMI_EN	SCC_SMI_STS
<b>Notes:</b> 1. SCI_EN must be 1 to enable SCI, except for BIOS_RLS. SCI_EN must be 0 to enable SMI. 2. SCI can be routed to cause interrupt 9:11 or 20:23 (20:23 only available in APIC mode). 3. GBL_SMI_EN must be 1 to enable SMI. 4. EOS must be written to 1 to re-enable SMI for the next 1. 5. The PCH must have SMI fully enabled when the PCH is also enabled to trap cycles. If SMI is not enabled in conjunction with the trap enabling, then hardware behavior is undefined. 6. When a power button override first occurs, the system will transition immediately to S5. The SCI will only occur after the next wake to S0 if the residual status bit (PRBTNOR_STS) is not cleared prior to setting SCI_EN. 7. GBL_STS being set will cause an SCI, even if the SCI_EN bit is not set. Software must take great care not to set the BIOS_RLS bit (which causes GBL_STS to be set) if the SCI handler is not in place. 8. Refer to <a href="#">General Purpose Input and Output (GPIO)</a> on page 127 for specific GPIOs enabled for SCIs and/or SMIs.				

### PCI Express\* SCI

PCI Express ports and the processor have the ability to cause PME using messages. When a PME message is received, the PCH will set the PCI\_EXP\_STS bit. If the PCI\_EXP\_EN bit is also set, the PCH can cause an SCI using the GPE0\_STS register.

### PCI Express\* Hot-Plug

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

## 25.4.5 C-States

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

## 25.4.6 Dynamic 24 MHz Clock Control

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

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

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

#### Conditions for Checking the 24 MHz Clock

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

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

- Cycles on LPC
- SERIRQ activity

#### Conditions for Maintaining the 24 MHz Clock

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

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

#### Conditions for Stopping the 24 MHz Clock

- When there is a lack of activity (as defined above) for 29 to 30 clock cycles, the PCH de-asserts (drive high) CLKRUN# for 1 clock cycle and then tri-states the signal.
- If no device drives CLKRUN# low within 6 clock cycles after it has been de-asserted, the PCH will stop the 24 MHz clocks.

#### Conditions for Re-starting the 24 MHz Clock

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

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

### 25.4.7 Sleep States

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

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

#### Initiating Sleep State

Sleep states (S3–S5) are initiated by:

- Masking interrupts, turning off all bus master enable bits, setting the desired type in the SLP\_TYP field, and then setting the SLP\_EN bit. The hardware then attempts to gracefully put the system into the corresponding Sleep state.
- Pressing the PWRBTN# Signal for more than 4 seconds to cause a Power Button Override event. In this case the transition to the S5 state is less graceful, since there are no dependencies on DMI messages from the processor or on clocks other than the RTC clock.
- Assertion of the THERMTRIP# signal will cause a transition to the S5 state. This can occur when system is in S0 state.
- Shutdown by integrated manageability functions (ASF/Intel® AMT).
- Internal watchdog timer Timeout events.

**Table 84. Sleep Types**

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

### Exiting Sleep States

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

Upon exit from the PCH-controlled Sleep states, the WAK\_STS bit is set. The possible causes of wake events (and their restrictions) are shown in below table.

### NOTE

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

**Table 85. Causes of Wake Events**

Cause	How Enabled	Wake from Sx	Wake from Deep Sx	Wake from Sx After Power Loss <sup>2</sup>	Wake from "Reset" Types <sup>3</sup>
RTC Alarm	Set RTC_EN bit in PM1_EN_STS register	Yes	Yes	Yes	No
Power Button	Always enabled as Wake event	Yes	Yes	Yes	Yes
Any GPIOs except DSW GPIOs can be enabled for wake <sup>5</sup>		Yes	No	No	No
<i>continued...</i>					

Cause	How Enabled	Wake from Sx	Wake from Deep Sx	Wake from Sx After Power Loss <sup>2</sup>	Wake from "Reset" Types <sup>3</sup>
LAN_WAKE#	Enabled natively (unless pin is configured to be in GPIO mode)	Yes	Yes	Yes	Yes
Intel® High Definition Audio	Event sets PME_B0_STS bit; PM_B0_EN must be enabled. Can not wake from S5 state if it was entered due to power failure or power button override	Yes	No	Yes	No
Primary PME#	PME_B0_EN bit in GPE0_EN[127:96] register	Yes	No	Yes	No
Secondary PME#	Set PME_EN bit in GPE0_EN[127:96] register	Yes	No	Yes	No
PCI Express* WAKE# pin	PCIEXP_WAKE_DIS bit	Yes	Yes	Yes	No
SMBALERT#	Refer to Note 4	Yes	No	Yes	Yes
SMBus Slave Wake Message (01h)	Wake/SMI# command always enabled as a Wake event <i>Note:</i> SMBus Slave Message can wake the system from S3–S5, as well as from S5 due to Power Button Override	Yes	No	Yes	Yes
SMBus Host Notify message received	HOST_NOTIFY_WKEN bit SMBus Slave Command register. Reported in the SMB_WAK_STS bit in the GPE0_STS register	Yes	No	Yes	Yes
Intel® CSME Non-Maskable Wake	Always enabled as a wake event	Yes	No	Yes	Yes
Integrated WoL Enable Override	WoL Enable Override bit (in Configuration Space)	Yes	Yes	Yes	Yes
Wake Alarm Device	WADT_EN in GPE0_EN[127:96]	Yes	Yes	No	No

**Notes:**

1. If BATLOW# signal is low, PCH will not attempt to wake from S3–S5 (nor will it exit Deep Sx), even if valid wake event occurs. This prevents the system from waking when battery power is insufficient to wake the system. However, once BATLOW# goes back high, the system will boot.
2. This column represents what the PCH would honor as wake events but there may be enabling dependencies on the device side which are not enabled after a power loss.
3. Reset Types include: Power Button override, Intel® CSME -initiated power button override, Intel® CSME -initiated host partition reset with power down, Intel® CSME Watchdog Timer, SMBus unconditional power down, processor thermal trip, PCH catastrophic temperature event.
4. SMBALERT# signal is multiplexed with a GPIO pin that defaults to GPIO mode. Hence, SMBALERT# related wakes are possible only when this GPIO is configured in native mode, which means that BIOS must program this GPIO to operate in native mode before this wake is possible. Because GPIO configuration is in the resume well, wakes remain possible until one of the following occurs: BIOS changes the pin to GPIO mode, a G3 occurs or Deep Sx entry occurs.
5. There are only 72 bits in the GPE registers to be assigned to GPIOs, though any of the GPIOs can trigger a wake, only those status of GPIO mapped to 1-tier scheme are directly accessible through the GPE status registers. For those GPIO mapped under 2-tier scheme, their status would be reflected under single master status, "GPIO\_TIER2\_SCI\_STS" or GPE0\_STS and further comparison needed to know which 2-tier GPI(s) has triggered the GPIO Tier 2 SCI.
6. A change in AC\_PRESENT causes an exit from Deep Sx to Sx, but the system will not wake all the way to S0.

### PCI Express\* WAKE# Signal and PME Event Message

PCI Express\* ports can wake the platform from any sleep state (S3, 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.

PCI Express\* ports and the processor have the ability to cause PME using messages. These are logically OR'd to set the single PCI\_EXP\_STS bit. When a PME message is received, the PCH will set the PCI\_EXP\_STS bit. If the PCI\_EXP\_EN bit is also set, the PCH can cause an SCI via GPE0\_STS register.

### Sx-G3-Sx, Handling Power Failures

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

The AFTERG3\_EN bit provides the ability to program whether or not the system should boot once power returns after a power loss event. If the policy is to not boot, the system remains in an S5 state (unless previously in S4). There are only three possible events that will wake the system after a power failure.

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

DSW\_PWROK going low would place the PCH into a G3 state.

Although PME\_EN is in the RTC well, this signal cannot wake the system after a power loss. PME\_EN is cleared by RTCRST#, and PME\_STS is cleared by RSMRST#.

**Table 86. Transitions Due to Power Failure**

State at Power Failure	AFTERG3_EN Bit	Transition when Power Returns
S0, S3	1 0	S5 S0
S4	1 0	S4 S0
S5	1 0	S5 S0
Deep S4	1 0	Deep S4 <sup>1</sup> S0
Deep S5	1 0	Deep S5 <sup>1</sup> 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 Suspend wells are powered off, while the Deep Sx Well (DSW) remains powered. A limited set of wake events are supported by the logic located in the DSW.

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

- **Entry Into Deep Sx**

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

All of the following must be met:

1. Intel® CSME in M-Off or M3-PG AND
2. Either a. or b. as defined below
  - a. (S4AC\_GATE\_SUS AND S4) OR (S5AC\_GATE\_SUS AND S5)), OR
  - b. ((AC\_PRESENT = 0) AND ((S3DC\_GATE\_SUS AND S3) OR (S4DC\_GATE\_SUS AND S4) OR (S5DC\_GATE\_SUS AND S5)))

---

**NOTES**

1. If REQ\_CNV\_NOWAKE\_DSX is set to '1,' connectivity wake must be disabled in addition to the above DeepSx entry conditions to allow DeepSx entry.
  2. If REQ\_BATLOW\_DSX is set to '1,' BATLOW# must be asserted in addition to the above DeepSx entry conditions to allow DeepSx entry.
- 

**Table 87. Supported Deep Sx Policy Configurations**

Configuration	S4DC_GATE_S US	S4AC_GATE_S US	S5DC_GATE_S US	S5AC_GATE_S US
1. Enabled in S5 when on Battery (ACPRESENT = 0)	0	0	1	0
2. Enabled in S5 (ACPRESENT not considered)	0	0	1	1
3. Enabled in S4 and S5 when on Battery (ACPRESENT = 0)	1	0	1	0
4. Enabled in S4 and S5 (ACPRESENT not considered)	1	1	1	1
5. Enabled in S4 and S5 when on Battery (ACPRESENT = 0)	1	0	1	0
6. Enabled in S4 and S5 (ACPRESENT not considered)	1	1	1	1
7. Deep S4/ S5 disabled	0	0	0	0
<i>Note:</i> All other configurations are RESERVED.				

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

- **Exit from Deep Sx**

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

**Table 88. Deep Sx Wake Events**

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

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

## 25.4.8 Event Input Signals and Their Usage

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

### PWRBTN# (Power Button)

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

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

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

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

**Table 89. 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 <i>Note:</i> Processing of transitions starts within 100 us of the PWRBTN# input pin to PCH going low. <sup>1</sup>
S3 – S5	PWRBTN# goes low	Wake Event. Transitions to S0 state	Standard wakeup <i>Note:</i> 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. <sup>1</sup>
Deep Sx	PWRBTN# goes low	Wake Event. Transitions to S0 state	Standard wakeup <i>Note:</i> Could be impacted by SLP_* min assertion. The minimum time the PWRBTN# pin should be asserted is 150 us. The PCH will start processing this change once the minimum time requirement is satisfied but subsequently the PWRBTN# pin needs to de-assert for at least 500 us after RSMRST# de-assertion otherwise the system waits indefinitely in S5 state. <sup>1</sup>
G3	PWRBTN# pressed	None	No effect since no power Not latched nor detected <i>Notes:</i> 1. During G3 exit, PWRBTN# pin must be kept de-asserted for a minimum time of 500 us after the RSMRST# has de-asserted. <sup>2</sup> 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. <sup>1</sup>
S0 – S4	PWRBTN# held low for at least 4 3 consecutive seconds	Unconditional transition to S5 state and if Deep Sx is enabled and conditions are met per <a href="#">Deep Sx</a> on page 197, the system will then transition to Deep Sx.	No dependence on processor or any other subsystem <i>Note:</i> 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.
<i>Notes:</i> 1. If PM_CFG.PB_DB_MODE='0', the debounce logic adds 16 ms to the start/minimum time for processing of power button assertions. 2. This minimum time is independent of the PM_CFG.PB_DB_MODE value. 3. The amount of time PWRBTN# must be asserted is configurable via PM_CFG2.PBOP. 4 seconds is the default.			

### Power Button Override Function

If PWRBTN# is observed active for at least four consecutive seconds (always sampled after the output from debounce logic), the PCH should unconditionally transition to the G2/S5 state or Deep Sx, regardless of present state (S0 – S4), even if the

PCH\_PWROK is not active. In this case, the transition to the G2/S5 state or Deep Sx does not depend on any particular response from the processor, nor any similar dependency from any other subsystem.

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

The PWRBTN# status is readable to check if the button is currently being pressed or has been released. If PM\_CFG.PB\_DB\_MODE='0', the status is taken after the de-bounce. If PM\_CFG.PB\_DB\_MODE='1', the status is taken before the de-bounce. In either case, the status is readable using the PWRBTN\_LVL bit.

---

**NOTE**

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

---

**Sleep Button**

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

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

**PME# (PCI Power Management Event)**

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

There is also an internal PME\_B0\_STS bit that will be set by the PCH when any internal device with PCI Power Management capabilities on bus 0 asserts the equivalent of the PME# signal. This is separate from the external PME# signal and can cause the same effect.

**SYS\_RESET# Signal**

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

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

---

**NOTES**

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

**THERMTRIP# Signal**

If THERMTRIP# goes active, the processor is indicating an overheat condition, and the PCH immediately transitions to an S5 state, driving SLP\_S3#, SLP\_S4#, SLP\_S5# low, and setting the GEN\_PMCN\_2.PTS bit. The transition will generally look like a power button override.

When a THERMTRIP# event occurs, the PCH will power down immediately without following the normal S0 -> S5 path. The PCH will immediately drive SLP\_S3#, SLP\_S4#, and SLP\_S5# low within 1 us after sampling THERMTRIP# active.

The reason the above is important is as follows:

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

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

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

---

**NOTE**

A thermal trip event will clear the PWRBTN\_STS bit.

---

**Sx\_Exit\_Holdoff#**

When S3/S4/S5 is entered and SLP\_A# is asserted, Sx\_Exit\_Holdoff# can be asserted by a platform component to delay resume to S0. SLP\_A# de-assertion is an indication of the intent to resume to S0, but this will be delayed so long as Sx\_Exit\_Holdoff# is asserted. Sx\_Exit\_Holdoff is ignored outside of an S3/S4/S5 entry sequence with SLP\_A# asserted. With the de-assertion of RSMRST# (either from G3->S0 or DeepSx->S0), this pin is a GPIO input and must be programmed by BIOS to operate as

Sx\_Exit\_Holdoff. When SLP\_A# is asserted (or it is de-asserted but Sx\_Exit\_Holdoff# is asserted), the PCH will not access SPI Flash. How a platform uses this signal is platform specific.

#### Requirements to support Sx\_Exit\_Holdoff#

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

After the PCH has booted up to S0 at least once since the last G3 or DeepSx exit, the EC can begin monitoring SLP\_A# and using the SX\_EXIT\_HOLDOFF# pin to stop the PCH from accessing flash. When SLP\_A# asserts, if the EC intends to access flash, it will assert SX\_EXIT\_HOLDOFF#. To cover the case where the PCH is going through a global reset, and not a graceful Sx+CMoff/Sx+CM3PG entry, the EC must monitor the SPI flash CS0# pin for 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.

### 25.4.9 ALT Access Mode

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

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

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

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

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

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

### Write Only Registers with Read Paths in ALT Access Mode

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

**Table 90. Write Only Registers with Read Paths in ALT Access Mode**

Restore Data			
I/O Addr	# of Rds	Access	Data
20h	12	1	PIC ICW2 of Master controller
		2	PIC ICW3 of Master controller
		3	PIC ICW4 of Master controller
		4	PIC OCW1 of Master controller <sup>1</sup>
		5	PIC OCW2 of Master controller
		6	PIC OCW3 of Master controller
		7	PIC ICW2 of Slave controller
		8	PIC ICW3 of Slave controller
		9	PIC ICW4 of Slave controller
		10	PIC OCW1 of Slave controller <sup>1</sup>
		11	PIC OCW2 of Slave controller
		12	PIC OCW3 of Slave controller
40h	7	1	Timer Counter 0 status, bits [5:0]
		2	Timer Counter 0 base count low byte
		3	Timer Counter 0 base count high byte
		6	Timer Counter 2 base count low byte
		7	Timer Counter 2 base count high byte
42h	1		Timer Counter 2 status, bits [5:0]
70h	1		Bit 7 = Read value is '0'. Bits [6:0] = RTC Address
Notes: 1. The OCW1 register must be read before entering ALT access mode. 2. Bits 5, 3, 1, and 0 return 0.			

### PIC Reserved Bits

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

**Table 91. PIC Reserved Bits Return Values**

PIC Reserved Bits	Value Returned
ICW2(2:0)	000
ICW4(7:5)	000
continued...	

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

## 25.4.10 System Power Supplies, Planes, and Signals

### Power Plane Control

The SLP\_S3# output signal can be used to cut power to the system core supply, since it only goes active for the Suspend-to-RAM state (typically mapped to ACPI S3). Power must be maintained to the PCH primary well, and to any other circuits that need to generate Wake signals from the Suspend-to-RAM state. During S3 (Suspend-to-RAM) all signals attached to powered down planes will be tri-stated or driven low, unless they are pulled using a Pull-up resistor.

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

The SLP\_S4# output signal is used to remove power to additional subsystems that are powered during SLP\_S3#, as well as power to the system memory, since the context of the system is saved on the disk. Cutting power to the memory may be done using the power supply, or by external FETs on the motherboard.

SLP\_S5# output signal can be used to cut power to the system core supply.

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

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

### SLP\_S4# and Suspend-to-RAM Sequencing

The system memory suspend voltage regulator is controlled by the Glue logic. The SLP\_S4# signal should be used to remove power to system memory rather than the SLP\_S5# signal. The SLP\_S4# logic in the PCH provides a mechanism to fully cycle the power to the DRAM and/or detect if the power is not cycled for a minimum time.

---

#### NOTE

To use the minimum DRAM power-down feature that is enabled by the SLP\_S4# Assertion Stretch Enable bit (D31:F0:A4h Bit 3), the DRAM power must be controlled by the SLP\_S4# signal.

---

### PCH\_PWROK Signal

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

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

---

#### NOTE

SYS\_RESET# is recommended for implementing the system reset button. This saves external logic that is needed if the PCH\_PWROK input is used. Additionally, it allows for better handling of the SMBus and processor resets and avoids improperly reporting power failures.

---

### BATLOW# (Battery Low)

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

### 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.

---

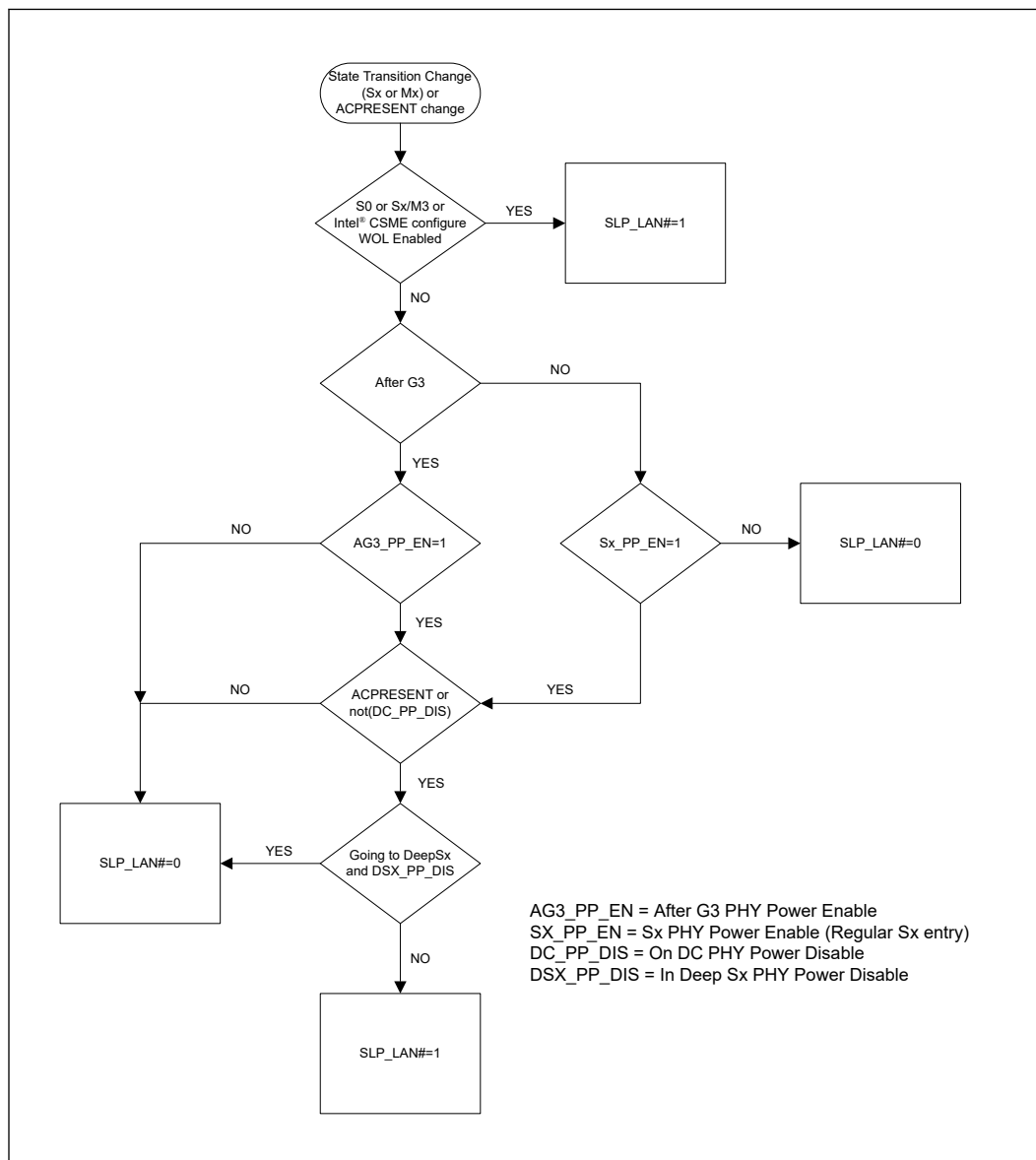
#### NOTE

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 35. 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 S3/S4/S5 states, the host BIOS must set HOST\_WLAN\_PP\_EN.
- If WoWLAN is required from Deep Sx, the host BIOS must set DSX\_WLAN\_PP\_EN.
- If Intel® CSME has access to the Wireless LAN device:

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

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

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 3.3V power rail.

### SUSPWRDNACK/SUSWARN#/GPP\_A13 Steady State Pin Behavior

Below table summarizes SUSPWRDNACK/SUSWARN#/GPP\_A13 pin behavior.

**Table 92. SUSPWRDNACK/SUSWARN#/GPP\_A13 Pin Behavior**

Pin	Deep Sx (Supported /Not-Supported)	GPP_A13 Input/Output (Determine by GP_IO_SEL bit)	Pin Value in S0	Pin Value in Sx/M-Off	Pin Value in Sx/M3	Pin Value in Deep Sx
SUSPWRDNACK	Not Supported	Native	0	Depends on Intel® CSME power package and power source <sup>1</sup>	0	OFF
SUSWARN#	Supported	Native	1	1 <sup>2</sup>	1	OFF
GPP_A13	Do not Care	IN	High-Z	High-Z	High-Z	OFF
	Do not Care	OUT	Depends on GPP_A13 output data value	Depends on GPP_A13 output data value	Depends on GPP_A13 output data value	OFF
<i>Notes:</i> 1. PCH will drive SPDA pin based on Intel® CSME power policy configuration. 2. If entering Deep Sx, pin will assert and become undriven ("Off") when suspend well drops upon Deep Sx entry.						

**Table 93. SUSPWRDNACK During Reset**

Reset Type (Note)	SPDA Value
Power-cycle Reset	0
Global Reset	0
Straight to S5	PCH initially drive '0' and then drive per Intel® CSME power policy configuration.
<i>Note:</i> Refer to <a href="#">Table 94</a> on page 210	

### RTCRST# and SRTCST#

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.

SRTCST# 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 SRTCST# not be pulled low in the S0 to S5 states.

## 25.4.11 Legacy Power Management Theory of Operation

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

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

### Mobile APM Power Management

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

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

## 25.4.12 Reset Behavior

When a reset is triggered, the PCH will send a warning message to the processor to allow the processor to attempt to complete any outstanding memory cycles and put memory into a safe state before the platform is reset. When the processor is ready, it will send an acknowledge message to the PCH. Once the message is received the PCH asserts PLTRST#.

The PCH does not require an acknowledge message from the processor to trigger PLTRST#. A global reset will occur after four seconds if an acknowledge from the processor is not received.

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

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

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

Straight to S5 is another reset type where all power wells that are controlled by the SLP\_S3#, SLP\_S4#, and SLP\_A# pins, as well as SLP\_S5# and SLP\_LAN# (if pins are not configured as GPIOs), are turned off. All PCH functionality is reset except RTC Power Well backed information and Suspend well status, configuration, and functional logic for controlling and reporting the reset. The host stays there until a valid wake event occurs.

The following table shows the various reset triggers.

**Table 94. Causes of Host and Global Resets**

Trigger	Host Reset Without Power Cycle <sup>1</sup>	Host Reset With Power Cycle <sup>2</sup>	Global Reset With Power Cycle <sup>3</sup>	Straight to S5 <sup>6</sup> (Host Stays There)
Write of 0Eh to CF9h (RST_CNT Register) when CF9h when Global Reset Bit=0b	No	Yes	No <sup>4</sup>	
Write of 06h to CF9h (RST_CNT Register) when CF9h when Global Reset Bit=0b	Yes	No	No <sup>4</sup>	
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 <sup>4</sup>	
SYS_RESET# Asserted and CF9h (RST_CNT Register) Bit 3 = 1	No	Yes	No <sup>4</sup>	
SMBus Slave Message received for Reset with Power-Cycle	No	Yes	No <sup>4</sup>	
SMBus Slave Message received for Reset without Power-Cycle	Yes	No	No <sup>4</sup>	
SMBus Slave Message received for unconditional Power Down	No	No	No	Yes
TCO Watchdog Timer reaches zero two times	Yes	No	No <sup>4</sup>	
Power Failure: PCH_PWROK signal goes inactive in S0 or DSW_PWROK drops	No	No	Yes	
SYS_PWROK Failure: SYS_PWROK signal goes inactive in S0	No	No	Yes	
Processor Thermal Trip (THERMTRIP#) causes transition to S5 and reset asserts	No	No	No	Yes
PCH internal thermal sensors signals a catastrophic temperature condition	No	No	No	Yes
<b>continued...</b>				

Trigger	Host Reset Without Power Cycle <sup>1</sup>	Host Reset With Power Cycle <sup>2</sup>	Global Reset With Power Cycle <sup>3</sup>	Straight to S5 <sup>6</sup> (Host Stays There)
Power Button 4 second override causes transition to S5 and reset asserts	N	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 <sup>4</sup>	
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 <sup>4</sup>	
Intel® Converged Security and Management Engine Triggered Host Reset without Power-Cycle	Yes	No	No <sup>4</sup>	
Intel® Converged Security and Management Engine Triggered Host Reset with Power-Cycle	No	Yes	No <sup>4</sup>	
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 <sup>8</sup>	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 <sup>5</sup>	No <sup>4</sup>	
PLTRST# Entry Timeout (Note 7)	No	No	Yes	
CPUPWRGD Stuck Low	No	No	Yes	
Power Management Watchdog Timer	No	No	No <sup>8</sup>	Yes
Intel® Converged Security and Management Engine Hardware Uncorrectable Error	No	No	No <sup>8</sup>	Yes

**Notes:**

1. The PCH drops this type of reset request if received while the system is in S3/S4/S5.
2. PCH does not drop this type of reset request if received while system is in a software-entered S3/S4/S5 state. However, the PCH will perform the reset without executing the RESET\_WARN protocol in these states.
3. The PCH does not send warning message to processor, reset occurs without delay.
4. Trigger will result in Global Reset with Power-Cycle if the acknowledge message is not received by the PCH.
5. The PCH waits for enabled wake event to complete reset.
6. Upon entry to S5, if Deep Sx is enabled and conditions are met, the system will transition to Deep Sx.
7. PLTRST# Entry Timeout is automatically initiated if the hardware detects that the PLTRST# sequence has not been completed within 4 seconds of being started.
8. Trigger will result in Global Reset with Power-Cycle if AGR\_LS\_EN=1 and Global Reset occurred while the current or destination state was S0.

## 26.0 Real Time Clock (RTC)

The PCH contains a Motorola\* MC146818B-compatible real-time clock with 256 bytes of battery-backed RAM. The real-time clock performs two key functions—keeping track of the time of day and storing system data, even when the system is powered down. The RTC operates on a 32.768 kHz crystal and a 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 30 days in advance, rather than just 24 hours in advance.

### Acronyms

Acronyms	Description
ESR	Equivalent Series Resistance. Resistive element in a circuit such as a clock crystal
GPI	General Purpose Input
PPM	Parts Per Million. Crystal accuracy or frequency variation indicator
RAM	Random Access Memory

## 26.1 Signal Description

Name	Type	Description
<b>RTCX1</b>	I	<b>Crystal Input 1:</b> This signal is connected to the 32.768 kHz crystal. If no external crystal is used, then RTCX1 can be driven with the desired clock rate. Maximum voltage allowed on this pin is 1.2V.
<b>RTCX2</b>	O	<b>Crystal Input 2:</b> This signal is connected to the 32.768 kHz crystal. If no external crystal is used, then RTCX2 must be left floating.
<b>RTCRST#</b>	I	<b>RTC Reset:</b> When asserted, this signal resets register bits in the RTC well. 1. Unless CMOS is being cleared (only to be done in the G3 power state) with a jumper, the RTCRST# input must always be high when all other RTC power planes are on. 2. In the case where the RTC battery is dead or missing on the platform, the RTCRST# pin must rise before the DSW_PWROK pin.
<b>SRTCRST#</b>	I	<b>Secondary RTC Reset:</b> This signal resets the manageability register bits in the RTC well when the RTC battery is removed. <i>Notes:</i> 1. The SRTCRST# input must always be high when all other RTC power planes are on. 2. In the case where the RTC battery is dead or missing on the platform, the SRTCRST# pin must rise before the DSW_PWROK pin. 3. SRTCRST# and RTCRST# should not be shorted together.

## 26.2 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>RTCRST#</b>	RTC	Undriven	Undriven	Undriven	Undriven
<b>SRTCRST#</b>	RTC	Undriven	Undriven	Undriven	Undriven

*Note:* 1. Reset reference for RTC well pins is RTCRST#.

## 26.3 Functional Description

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

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

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

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

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

### NOTES

1. The leap year determination for adding a 29th day to February does not take into account the end-of-the-century exceptions. The logic simply assumes that all years divisible by 4 are leap years. According to the Royal Observatory Greenwich, years that are divisible by 100 are typically not leap years. In every fourth century (years divisible by 400, like 2000), the 100-year-exception is over-ridden and a leap-year occurs.
2. The year 2100 will be the first time in which the current RTC implementation would incorrectly calculate the leap-year. The PCH does not implement month/year alarms.

### 26.3.1 Update Cycles

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

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

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

---

#### WARNING

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

---

### 26.3.2 Interrupts

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

### 26.3.3 Lockable RAM Ranges

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

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

### 26.3.4 Century Rollover

The PCH detects a rollover when the Year byte transitions from 99 to 00. Upon detecting the rollover, the PCH sets the NEWCENTURY\_STS bit.

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

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

### 26.3.5 Clearing Battery-Backed RTC RAM

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

#### Using RTCRST# to Clear CMOS

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

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

#### Using a GPI to Clear CMOS

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

#### NOTE

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

#### WARNING

Do not implement a jumper on VCCRTC to clear CMOS.

### 26.3.6 External RTC Circuitry

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

**Table 95. RTC Crystal Requirements**

Parameter	Specification
Frequency	32.768 kHz
Typical Tolerance	20 ppm or better
ESR	< 50 kOhm

**Table 96. External Crystal Oscillator Requirements**

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

## 27.0 Serial ATA (SATA)

The PCH SATA controller support two modes of operation, AHCI mode using memory space and RAID mode. The PCH SATA controller no longer supports IDE legacy mode using I/O space. Therefore, AHCI software is required. The PCH SATA controller supports the Serial ATA Specification, Revision 3.2.

### 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

### References

Specification	Location
Serial ATA Specification, Revision 3.2	<a href="https://www.sata-io.org">https://www.sata-io.org</a>
Serial ATA II: Extensions to Serial ATA 1.0, Revision 1.0	<a href="https://www.sata-io.org">https://www.sata-io.org</a>
Serial ATA II Cables and Connectors Volume 2 Gold	<a href="https://www.sata-io.org">https://www.sata-io.org</a>
Advanced Host Controller Interface Specification	<a href="http://www.intel.com/content/www/us/en/io/serial-ata/ahci.html">http://www.intel.com/content/www/us/en/io/serial-ata/ahci.html</a>

## 27.1 Signals Description

Name	Type	Description
<b>SATA_DEVSLP0 / GPP_E4</b>	OD	<p><b>Serial ATA Port [0] Device Sleep:</b> 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.</p> <p>Design Constraint: No external Pull-up or Pull-down termination required when used as DEVSLP.</p> <p><i>Note:</i> This pin can be mapped to SATA Port 0.</p>
<b>SATA_DEVSLP1 / GPP_E5</b>	OD	<p><b>Serial ATA Port [1] Device Sleep:</b> 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.</p> <p>Design Constraint: No external Pull-up or Pull-down termination required when used as DEVSLP.</p> <p><i>Note:</i> This pin can be mapped to SATA Port 1.</p>
<b>continued...</b>		

Name	Type	Description
<b>SATA_DEVSLP2</b> / GPP_E6	OD	<b>Serial ATA Port [2] Device Sleep:</b> 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. <i>Note:</i> This pin can be mapped to SATA Port 2.
<b>SATA_DEVSLP3</b> / GPP_F5	OD	<b>Serial ATA Port [3] Device Sleep:</b> 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. <i>Note:</i> This pin can be mapped to SATA Port 3.
<b>SATA_DEVSLP4</b> / GPP_F6	OD	<b>Serial ATA Port [4] Device Sleep:</b> 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. <i>Note:</i> This pin can be mapped to SATA Port 4.
<b>SATA_DEVSLP5</b> / GPP_F7	OD	<b>Serial ATA Port [5] Device Sleep:</b> 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. <i>Note:</i> This pin can be mapped to SATA Port 5.
<b>SATA0A_TXP</b> / PCIE11_TXP <b>SATA0A_TXN</b> / PCIE11_TXN	O	<b>Serial ATA Differential Transmit Pair 0 [First Instance]:</b> These outbound SATA Port 0 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 11 signals.
<b>SATA0A_RXP</b> / PCIE11_RXP <b>SATA0A_RXN</b> / PCIE11_RXN	I	<b>Serial ATA Differential Receive Pair 0 [First Instance]:</b> These inbound SATA Port 0 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 11 signals.
<b>SATA1A_TXP</b> / PCIE12_TXP <b>SATA1A_TXN</b> / PCIE12_TXN	O	<b>Serial ATA Differential Transmit Pair 1 [First Instance]:</b> These outbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 12.
<b>SATA1A_RXP</b> / PCIE12_RXP <b>SATA1A_RXN</b> / PCIE12_RXN	I	<b>Serial ATA Differential Receive Pair 1 [First Instance]:</b> These inbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 12.
<b>SATA0B_TXP</b> / PCIE13_TXP <b>SATA0B_TXN</b> / PCIE13_TXN	O	<b>Serial ATA Differential Transmit Pair 0 [Second Instance]:</b> These outbound SATA Port 0 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 13 signals.
<b>SATA0B_RXP</b> / PCIE13_RXP <b>SATA0B_RXN</b> / PCIE13_RXN	I	<b>Serial ATA Differential Receive Pair 0 [Second Instance]:</b> These inbound SATA Port 0 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 13 signals.
<b>SATA1B_TXP</b> / PCIE14_TXP	O	<b>Serial ATA Differential Transmit Pair 1 [Second Instance]:</b> These outbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 11 signals.

continued...

Name	Type	Description
<b>SATA1B_TXN</b> / PCIE14_TXN		
<b>SATA1B_RXP</b> / PCIE14_RXP <b>SATA1B_RXN</b> / PCIE14_RXN	I	<b>Serial ATA Differential Receive Pair 1 [Second Instance]:</b> These inbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 11 signals.
<b>SATA2_TXP</b> / PCIE15_TXP <b>SATA2_TXN</b> / PCIE15_TXN	O	<b>Serial ATA Differential Transmit Pair 2:</b> These outbound SATA Port 2 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 15 signals.
<b>SATA2_RXP</b> / PCIE15_RXP <b>SATA2_RXN</b> / PCIE15_RXN	I	<b>Serial ATA Differential Receive Pair 2:</b> These inbound SATA Port 2 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 15 signals.
<b>SATA3_TXP</b> / PCIE16_TXP <b>SATA3_TXN</b> / PCIE16_TXN	O	<b>Serial ATA Differential Transmit Pair 3:</b> These outbound SATA Port 3 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 16 signals.
<b>SATA3_RXP</b> / PCIE16_RXP <b>SATA3_RXN</b> / PCIE16_RXN	I	<b>Serial ATA Differential Receive Pair 3:</b> These inbound SATA Port 3 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 16 signals.
<b>SATA4_TXP</b> / PCIE17_TXP <b>SATA4_TXN</b> / PCIE17_TXN	O	<b>Serial ATA Differential Transmit Pair 4:</b> These outbound SATA Port 4 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 17 signals.
<b>SATA4_RXP</b> / PCIE17_RXP <b>SATA4_RXN</b> / PCIE17_RXN	I	<b>Serial ATA Differential Receive Pair 4:</b> These inbound SATA Port 4 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 17 signals.
<b>SATA5_TXP</b> / PCIE18_TXP <b>SATA5_TXN</b> / PCIE18_TXN	O	<b>Serial ATA Differential Transmit Pair 5:</b> These outbound SATA Port 5 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 18 signals.
<b>SATA5_RXP</b> / PCIE18_RXP <b>SATA5_RXN</b> / PCIE18_RXN	I	<b>Serial ATA Differential Receive Pair 5:</b> These inbound SATA Port 5 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s. The signals are multiplexed with PCIe* Port 18 signals.
<b>SATAGP0</b> / GPP_E0 / SATAXPCIE0	I	<b>Serial ATA Port [0] General Purpose Inputs:</b> 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. <i>Note:</i> The default use of this pin is GPP_E0. Pin defaults to Native mode as SATAXPCIE0 depends on soft-strap.
<b>SATAGP1</b> / GPP_E1 / SATAXPCIE1	I	<b>Serial ATA Port [1] General Purpose Inputs:</b> 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. <i>Note:</i> This default use of this pin is GPP_E1. Pin defaults to Native mode as SATAXPCIE1 depends on soft-strap.
<b>SATAGP2</b> / GPP_E2 / SATAXPCIE2	I	<b>Serial ATA Port [2] General Purpose Inputs:</b> 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.
<b>continued...</b>		

Name	Type	Description
		<i>Note:</i> The default use of this pin is GPP_E2. Pin defaults to Native mode as SATAXPCIE2 depends on soft-strap.
<b>SATAGP3</b> / SATAXPCIE3 / GPP_F0	I	<b>Serial ATA Port [3] General Purpose Inputs:</b> 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. <i>Note:</i> The default use of this pin is GPP_F0. Pin defaults to Native mode as SATAXPCIE3 depends on soft-strap.
<b>SATAGP4</b> / SATAXPCIE4 / GPP_F1	I	<b>Serial ATA Port [4] General Purpose Inputs:</b> 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. <i>Note:</i> The default use of this pin is GPP_F1. Pin defaults to Native mode as SATAXPCIE4 depends on soft-strap.
<b>SATAGP5</b> / SATAXPCIE5 / GPP_F2	I	<b>Serial ATA Port [5] General Purpose Inputs:</b> When configured as SATAGP5, this is an input pin that is used as an interlock switch status indicator for SATA Port 5. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open. <i>Note:</i> The default use of this pin is GPP_F2. Pin defaults to Native mode as SATAXPCIE5 depends on soft-strap.
<b>SATALED#</b> / GPP_E8 / SPI1_CS1#	OD O	<b>Serial ATA LED:</b> 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. <i>Note:</i> An external Pull-up resistor to VCC3_3 is required.
<b>SATA_SCLOCK</b> / GPP_F10	OD	<b>SGPIO Reference Clock:</b> The SATA controller uses rising edges of this clock to transmit serial data, and the target uses the falling edge of this clock to latch data. The SClock frequency supported is 32 kHz. If SGPIO interface is not used, this signal can be used as GPP_F10.
<b>SATA_SLOAD</b> / GPP_F11	OD	<b>SGPIO Load:</b> The controller drives a '1' at the rising edge of SCLOCK to indicate either the start or end of a bit stream. A 4-bit vendor specific pattern will be transmitted right after the signal assertion. If SGPIO interface is not used, this signal can be used as GPP_F11.
<b>SATA_SDATAOUT0</b> / GPP_F13	OD	<b>SGPIO Dataout0:</b> Driven by the controller to indicate the drive status in the following sequence: drive 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2... If SGPIO interface is not used, the signals can be used as GPP_F13.
<b>SATA_SDATAOUT1</b> / GPP_F12	OD	<b>SGPIO Dataout1:</b> Driven by the controller to indicate the drive status in the following sequence: drive 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2... If SGPIO interface is not used, the signals can be used as GPP_F12.

## 27.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor type	Notes
SATAXPCIE[7:0]	Internal pull-up	Internal Pull-Up Resistors are 14 -26 kohm unless specified.

## 27.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>4</sup>	Immediately after Reset <sup>4</sup>	S3/S4/S5	Deep Sx
SATA0A_TXP/N, SATA0A_RXP/N	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SATA0B_TXP/N, SATA0B_RXP/N	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SATA1A_TXP/N, SATA1A_RXP/N	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SATA1B_TXP/N, SATA1B_RXP/N	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SATA[7:2]_TXP/N, SATA[7:2]_RXP/N <sup>3</sup>	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SATALED#/GPP_E81	Primary	Undriven	Undriven	Undriven	OFF
DEVSLP[2:0]/ GPP_E[6:4] <sup>1</sup>	Primary	Undriven	Undriven	Driven Low	OFF
DEVSLP[7:3]/ GPP_F[9:5] <sup>1,3</sup>	Primary	Undriven	Undriven	Driven Low	OFF
SATAGP[2:0]/ GPP_E[2:0] <sup>2</sup>	Primary	Undriven	Undriven	Undriven	OFF
SATAGP[7:3]/ GPP_F[4:0] <sup>2,3</sup>	Primary	Undriven	Undriven	Undriven	OFF
SATAXPCIE[7:0] <sup>2,3</sup>	Primary	Internal Pull-up	Internal Pull-up	Undriven	OFF
SATA_SCLOCK/GPP_F10 <sup>1</sup>	Primary	Undriven	Undriven	Undriven	OFF
SATA_SLOAD/GPP_F11 <sup>1</sup>	Primary	Undriven	Undriven	Undriven	OFF
SATA_SDATAOUT0/ GPP_F13 <sup>1</sup>	Primary	Undriven	Undriven	Undriven	OFF
SATA_SDATAOUT1/ GPP_F12 <sup>1</sup>	Primary	Undriven	Undriven	Undriven	OFF
<p><b>Notes:</b> 1. Pin defaults to GPIO mode. The pin state during and immediately after reset follows default GPIO mode pin state. The pin state for S0 to Deep Sx reflects assumption that GPIO Use Select register was programmed to native mode functionality. If GPIO Use Select register is programmed to GPIO mode, refer to Multiplexed GPIO (Defaults to GPIO Mode) section for the respective pin states in S0 to Deep Sx.</p> <p>2. Pin defaults to Native mode as SATAXPCIE depends on soft-strap.</p> <p>3. Reset reference for primary well pins is RSMRST#.</p>					

## 27.4 Functional Description

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

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

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

### 27.4.1 SATA 6 Gb/s Support

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

### 27.4.2 SATA Feature Support

The PCH SATA controller is capable of supporting all AHCI 1.3 and AHCI 1.3.1, refer to the Intel web site on Advanced Host Controller Interface Specification for current specification status: <http://www.intel.com/content/www/us/en/io/serial-ata/ahci.html>.

For capability details, refer to PCH SATA controller register (D23:F0:Offset 00h CAP, and AHCI BAR PxCMD Offset 18h).

The PCH SATA controller does **not** support:

- Port Multiplier
- FIS Based Switching
- Command Based Switching
- IDE mode or combination mode
- Cold Presence Detect
- Function Level Reset (FLR)

### 27.4.3 Hot-Plug Operation

The PCH SATA controller supports Hot-Plug Surprise removal and Insertion Notification. An internal SATA port with a Mechanical Presence Switch can support PARTIAL and SLUMBER with Hot-Plug Enabled. Software can take advantage of power savings in the low power states while enabling Hot-Plug operation. Refer to [Thermal Management](#) on page 48 of the AHCI specification for details.

### 27.4.4 Intel® Rapid Storage Technology (Intel® RST)

The PCH SATA controller provides support for Intel® Rapid Storage Technology, providing both AHCI and integrated RAID functionality. The RAID capability provides high-performance/data-redundancy RAID 0/1/5/10 functionality on up to six ports for the PCH of the PCH SATA controller. Matrix RAID support is provided to allow multiple RAID levels to be combined on a single set of hard drives, such as RAID 0 and RAID 1 on two disks. Other RAID features include hot spare support, SMART alerting, and RAID 0 auto replace. Software components include an Option ROM and UEFI Driver for pre-boot configuration and boot functionality, a Windows\* compatible driver, and a user interface for configuration and management of the RAID capability of PCH 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 PCH SATA controller.

- RAID Level 0 performance scaling up to 6 drives, enabling higher throughput for data intensive applications such as video editing.

- Data redundancy is offered through RAID Level 1, which performs mirroring.
- RAID Level 10 provides high levels of storage performance with data protection, combining the fault-tolerance of RAID Level 1 with the performance of RAID Level 0. By striping RAID Level 1 segments, high I/O rates can be achieved on systems that require both performance and fault-tolerance. RAID Level 10 requires 4 hard drives, and provides the capacity of two drives.
- RAID Level 5 provides highly efficient storage while maintaining fault-tolerance on 3 or more drives. By striping parity, and rotating it across all disks, fault tolerance of any single drive is achieved while only consuming 1 drive worth of capacity. That is, a 3-drive RAID 5 has the capacity of 2 drives, or a 4-drive RAID 5 has the capacity of 3 drives. RAID 5 has high read transaction rates, with a medium write rate. RAID 5 is well suited for applications that require high amounts of storage while maintaining fault tolerance.

By using the PCH's built-in Intel® Rapid Storage Technology, there is no loss of additional PCIe\*/system resources or add-in card slot/motherboard space footprint used compared to when a discrete RAID controller is implemented. Intel® Rapid Storage Technology functionality requires the following items:

1. PCH SKU enabled for Intel® Rapid Storage Technology.
2. Intel® Rapid Storage Technology RAID Option ROM or UEFI Driver must be on the platform.
3. Intel® Rapid Storage Technology drivers, most recent revision.
4. At least two SATA hard disk drives (minimum depends on RAID configuration).

Intel® Rapid Storage Technology is not available in the following configurations:

1. The SATA controller is programmed in RAID mode, but the AIE bit (D23:F0:Offset 9Ch bit 7) is set to 1.

### **Intel® Rapid Storage Technology (Intel® RST) RAID Option ROM**

The Intel® Rapid Storage Technology RAID Option ROM is a standard PnP Option ROM that is easily integrated into any System BIOS. When in place, it provides the following three primary functions:

- Provides a text mode user interface that allows the user to manage the RAID configuration on the system in a pre-operating system environment. Its feature set is kept simple to keep size to a minimum, but allows the user to create and delete RAID volumes and select recovery options when problems occur.
- Provides boot support when using a RAID volume as a boot disk. It does this by providing Int13 services when a RAID volume needs to be accessed by MS-DOS applications (such as NTLDR) and by exporting the RAID volumes to the System BIOS for selection in the boot order.
- At each boot up, provides the user with a status of the RAID volumes and the option to enter the user interface by pressing CTRL-I.

## **27.4.5 Power Management Operation**

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

## Power State Mappings

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

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

- **D0** – Device is working and instantly available.
- **D1** – Device enters when it receives a STANDBY IMMEDIATE command. Exit latency from this state is in seconds.
- **D3** – From the SATA device's perspective, no different than a D1 state, in that it is entered using the STANDBY IMMEDIATE command. However, an ACPI method is also called which will reset the device and then cut its power.

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

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

- **PHY READY** – PHY logic and PLL are both on and in active state.
- **Partial** – PHY logic is powered up, and in a reduced power state. The link PM exit latency to active state maximum is 10 ns.
- **Slumber** – PHY logic is powered up, and in a reduced power state. The link PM exit latency to active state maximum is 10 ms.
- **DevsIp** – 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.

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.

---

### • DevsIp 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 most likely to be used in ATA/ATAPI is the “STANDBY IMMEDIATE” command.

- **Host Controller D3<sub>HOT</sub> State**

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

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

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

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

### Low Power Platform Consideration

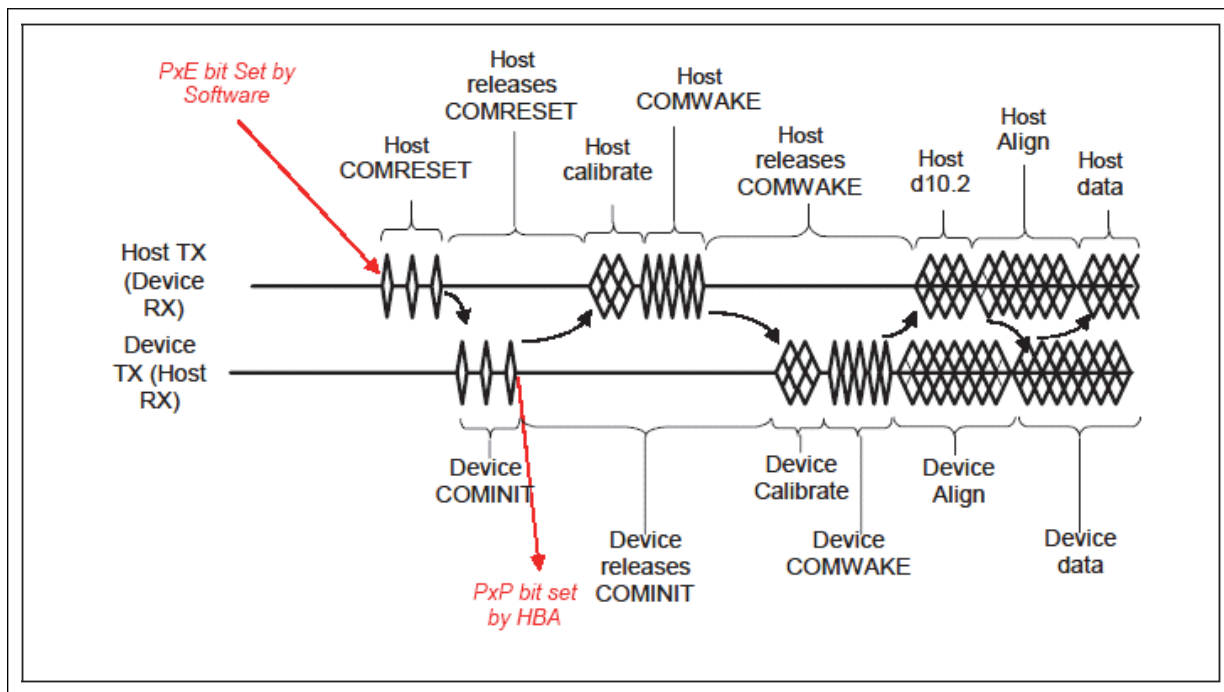
When low power feature is enabled, the Intel SATA controller may power off PLLs or OOB detection circuitry while in the Slumber link power state. As a result, a device initiated wake may not be recognized by the host. For example, when the low power feature is enabled it can prevent a Zero Power ODD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

The SATA MPHY Dynamic Power Gating (PHYDPGEPx) can be enabled/disabled for each SATA ports. Refer to SATA SIR Index 90h for the PHYDPGEPx register details.

### 27.4.6 SATA Device Presence

The flow used to indicate SATA device presence is shown in below figure. The 'PxP' bit refers to PCS.P[7:0]E bits, depending on the port being checked and the 'PxP' bits refer to the PCS.P[2:0]P bits, depending on the port being checked. If the PCS/PxP bit is set a device is present, if the bit is cleared a device is not present. If a port is disabled, software can check to see if a new device is connected by periodically re-enabling the port and observing if a device is present, if a device is not present it can disable the port and check again later. If a port remains enabled, software can periodically poll PCS.PxP to see if a new device is connected.

**Figure 36. Flow for Port Enable/Device Present Bits**



### 27.4.7 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.

### 27.4.8 Advanced Host Controller Interface (AHCI) Operation

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

AHCI defines transactions between the SATA controller and software and enables advanced performance and usability with SATA. Platforms supporting AHCI may take advantage of performance features such as no master/slave designation for SATA devices—each device is treated as a master—and hardware assisted native command

queuing. AHCI also provides usability enhancements such as hot-plug and advanced power management. AHCI requires appropriate software support (such as, an AHCI driver) and for some features, hardware support in the SATA device or additional platform hardware. Visit the Intel web site for current information on the AHCI specification.

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

---

**NOTE**

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

---

### 27.4.9 Enclosure Management (SGPIO Signals)

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

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

---

**NOTE**

These signals are not related to SATALED#.

---

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

---

**NOTE**

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

---

#### Mechanism

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

Software creates messages for transmission in the enclosure management message buffer. The data in the message buffer should not be changed if CTL.TM bit is set by software to transmit an update message. Software should only update the message

buffer when CTL.TM bit is cleared by hardware otherwise the message transmitted will be indeterminate. Software then writes a register to cause hardware to transmit the message or take appropriate action based on the message content. The software should only create message types supported by the controller, which is LED messages for the PCH. If the software creates other non LED message types (such as, SAF-TE, SES-2), the SGPIO interface may hang and the result is indeterminate.

During reset all SGPIO pins will be in tri-state. The interface will continue staying in tri-state after reset until the first transmission occurs, when software programs the message buffer and sets the transmit bit CTL.TM. The SATA host controller will initiate the transmission by driving SCLOCK and at the same time driving the SLOAD to "0" prior to the actual bit stream transmission. The Host will drive SLOAD low for at least 5 SCLOCK then only start the bit stream by driving the SLOAD to high. SLOAD will be driven high for 1 SCLOCK, followed by vendor-specific pattern that is default to "0000" if software is yet to program the value. A total of 24-bit streams from 8 ports (Port 0, Port 1, Port 2, Port 3, Port 4, Port 5, Port 6, Port 7) of 3-bit per port LED message will be transmitted on SDATAOUT0 pin after the SLOAD is driven high for 1 SCLOCK. For 8 SATA port configuration, only 4 ports (port 4, port 5, port 6 and port 7) of 12 bit total LED message follow by 12 bits of tri-state value will be transmitted out on SDATAOUT1 pin. For 6 SATA port configuration, only 2 ports (port 4 and port 5) of 6 bit total LED message follow by 18 bits of tri-state value will be transmitted out on SDATAOUT1 pin. For 4 SATA port configuration, SDATAOUT1 pin is not required hence can be tri-state always.

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

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

---

**NOTE**

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

---

**Message Format**

Messages shall be constructed with a one DWord header that describes the message to be sent followed by the actual message contents. The first DWord shall be constructed as shown in Enclosure Management Message Format (EM\_MF) register, refer to PCH Datasheet Volume 2.

The SAF-TE, SES-2, and SGPIO message formats are defined in the corresponding specifications, respectively. The LED message type is defined in the Enclosure Management LED (EM\_LED) register, refer to PCH Datasheet Volume 2. It is the responsibility of software to ensure the content of the message format is correct. If

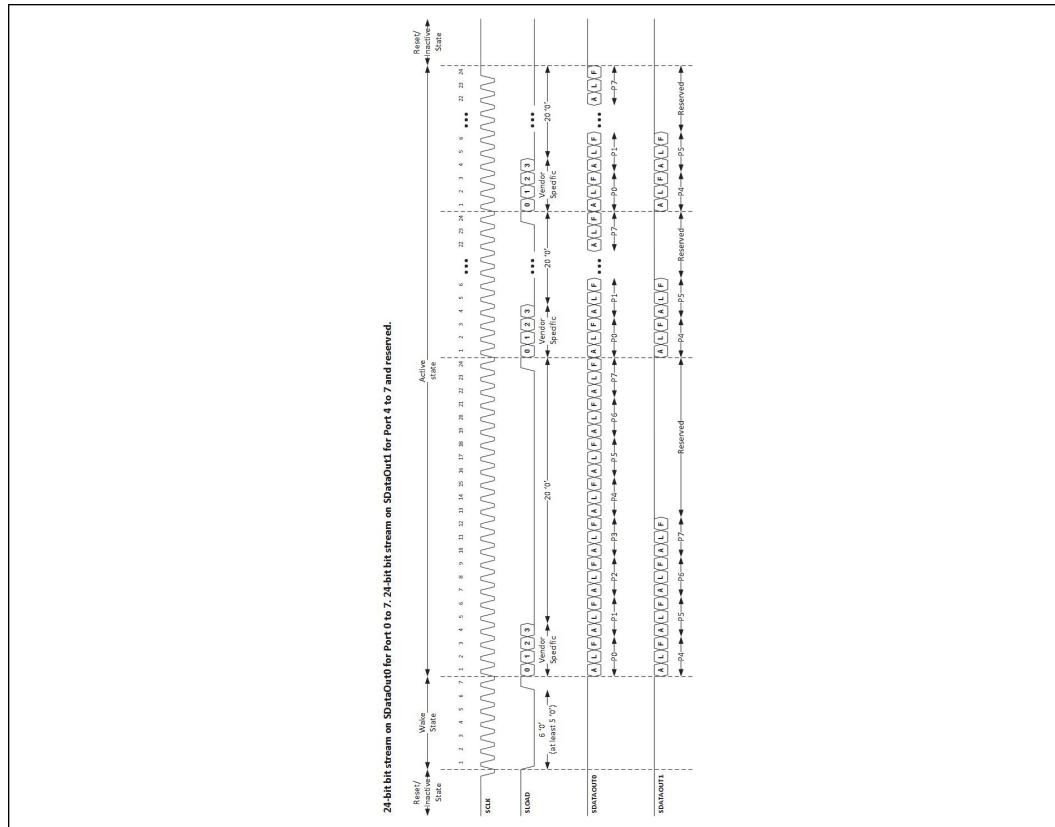
the message type is not programmed as 'LED' for this controller, the controller shall not take any action to update its LEDs. For LED message type, the message size always consists of 4 bytes.

### LED Message Type

The LED message type specifies the status of up to three LEDs. Typically, the usage for these LEDs is activity, fault, and locate. Not all implementations necessarily contain all LEDs (for example, some implementations may not have a locate LED). The message identifies the HBA port number and the Port Multiplier port number that the slot status applies to. If a Port Multiplier is not in use with a particular device, the Port Multiplier port number shall be '0'. The format of the LED message type is defined in the Enclosure Management LED (EM\_LED) register, refer to PCH Datasheet Volume 2. The LEDs shall retain their values until there is a following update for that particular slot.

### SGPIO Waveform

**Figure 37. Serial Data transmitted over the SGPIO Interface**



## 28.0 System Management Interface and SMLink

The PCH provides two SMLink interfaces, SMLink0 and SMLink1. The interfaces are intended for system management and are controlled by the Intel® CSME. Refer to [System Management](#) on page 38 for more detail.

### Acronyms

Acronyms	Description
BMC	Baseboard Management Controller
EC	Embedded Controller

### 28.1 Signal Description

Name	Type	Description
<b>INTRUDER#</b>	I	<b>Intruder Detect:</b> This signal can be set to disable the system if box detected open.
<b>SML0DATA/</b> GPP_C4	I/OD	<b>System Management Link 0 Data:</b> SMBus link to external PHY. External pull-up resistor is required.
<b>SML0CLK /</b> GPP_C3	I/OD	<b>System Management Link 0 Clock</b> External pull-up resistor is required.
<b>SML0ALERT# /</b> GPP_C5	I/OD	<b>System Management 0 Alert:</b> Alert for the SMBus controller to optional Embedded Controller or BMC. External pull-up resistor is required.
<b>SML1CLK /</b> GPP_C6	I/OD	<b>System Management Link 1 Clock:</b> SMBus link to optional Embedded Controller or BMC. External pull-up resistor is required.
<b>SML1DATA/</b> GPP_C7	I/OD	<b>System Management Link 1 Data:</b> SMBus link to optional Embedded Controller or BMC. External pull-up resistor is required.
<b>SML1ALERT# /</b> PCHHOT# / GPP_B23	I/OD	<b>System Management 1 Alert:</b> Alert for the SMBus controller to optional Embedded Controller or BMC. A soft-strap determines the native function SML1ALERT# or PCHHOT# usage. External pull-up resistor is required.
<b>PCHHOT# /</b> GPP_B23 / SML1ALERT#	OD	<b>PCHHOT#:</b> This signal is used to indicate a PCH temperature out of bounds condition to an external EC. External pull-up resistor is required.
<b>SML4DATA/</b> GPP_H17	I/OD	<b>System Management Link 4 Data:</b> External pull-up resistor is required.

### 28.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value
SML[2:0]ALERT#	Pull-down	20 kOhm $\pm$ 30%
PCHHOT#	Pull-down	20 kOhm $\pm$ 30%

## 28.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
INTRUDER#	RTC	Undriven	Undriven	Undriven	Undriven
SML[4:0]DATA	Primary	Undriven	Undriven	Undriven	OFF
SML[4:0]CLK	Primary	Undriven	Undriven	Undriven	OFF
SML[2:0]ALERT#	Primary	Pull-down (Internal)	Driven Low	Driven Low	OFF
SML[4:3]ALERT#	Primary	Undriven	Undriven	Undriven	OFF
PCHHOT#	Primary	Pull-down (Internal)	Driven Low	Pull-down (Internal)	OFF
<i>Note:</i> 1. Reset reference for primary well pins is RSMRST# and RTC well pins is RTCRST#.					

## 28.4 Functional Description

The SMLink interfaces are controlled by the Intel® CSME.

SMLink0 is mainly used for integrated LAN. When an Intel LAN PHY is connected to SMLink0, a soft strap must be set to indicate that the PHY is connected to SMLink0. The interface will be running at a 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.

## 29.0 Host System Management Bus (SMBus) Controller

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

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

### Acronyms

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

### References

Specification	Location
System Management Bus (SMBus) Specification, Version 2.0	<a href="http://www.smbus.org/specs/">http://www.smbus.org/specs/</a>

## 29.1 Signal Description

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

## 29.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
<b>SMBALERT#</b>	Pull-down	20 kohm ± 30%	The integrated pull down is disabled after PCH_PWROK assertion

## 29.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>SMBDATA</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>SMBCLK</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>SMBALERT#</b>	Primary	Pull-down (Internal)	Driven Low	Driven Low	OFF
<i>Note:</i> 1. Reset reference for primary well pins is RSMRST#, DSW well pins is DSW_PWROK, and RTC well pins is RTCRST#.					

## 29.4 Functional Description

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

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

### 29.4.1 Host Controller

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

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

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

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

#### Host Controller Operation Overview

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

The host controller supports 8 command protocols of the SMBus interface (Refer to System Management Bus (SMBus) Specification, Version 2.0): Quick Command, Send Byte, Receive Byte, Write Byte/Word, Read Byte/Word, Process Call, Block Read/Write, and Block Write-Block Read Process Call.

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

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

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

### Command Protocols

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

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

### Quick Command

When programmed for a Quick Command, the Transmit Slave Address Register is sent. The PEC byte is never appended to the Quick Protocol. Software should force the PEC\_EN bit to 0 when performing the Quick Command. Software must force the I2C\_EN bit to 0 when running this command. Refer to *System Management Bus (SMBus) Specification*, Version 2.0 for the format of the protocol.

### Send Byte/Receive Byte

For the Send Byte command, the Transmit Slave Address and Device Command Registers are sent. For the Receive Byte command, the Transmit Slave Address Register is sent. The data received is stored in the DATA0 register. Software must force the I2C\_EN bit to 0 when running this command.

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

### Write Byte/Word

The first byte of a Write Byte/Word access is the command code. The next 1 or 2 bytes are the data to be written. When programmed for a Write Byte/Word command, the Transmit Slave Address, Device Command, and Data0 Registers are sent. In addition, the Data1 Register is sent on a Write Word command. Software must force the I2C\_EN bit to 0 when running this command. Refer to *System Management Bus (SMBus) Specification*, Version 2.0 for the format of the protocol.

### Read Byte/Word

Reading data is slightly more complicated than writing data. First the PCH must write a command to the slave device. Then it must follow that command with a repeated start condition to denote a read from that device's address. The slave then returns 1 or 2 bytes of data. Software must force the I2C\_EN bit to 0 when running this command.

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

### Process Call

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

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

The Process Call command with I2C\_EN set and the PEC\_EN bit set produces undefined results. Software must force either I2C\_EN or PEC\_EN to 0 when running this command. Refer to *System Management Bus (SMBus) Specification, Version 2.0* for the format of the protocol.

---

#### NOTE

For process call command, the value written into bit 0 of the Transmit Slave Address Register needs to be 0.

---

---

#### NOTE

If the I2C\_EN bit is set, the protocol sequence changes slightly, the Command Code (Bits 18:11 in the bit sequence) are not sent. As a result, the slave will not acknowledge (Bit 19 in the sequence).

---

### Block Read/Write

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

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

For a Block Write, software must either force the I2C\_EN bit or both the PEC\_EN and AAC bits to 0 when running this command.

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

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

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

#### NOTE

For Block Write, if the I<sup>2</sup>C\_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<sup>2</sup>C mode (I<sup>2</sup>C\_EN bit is set), the PCH will never use the 32-byte buffer for any block commands.

#### I<sup>2</sup>C Read

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

However, this does not allow access to devices using the I<sup>2</sup>C “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 I<sup>2</sup>C\_EN bit. The I<sup>2</sup>C 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<sup>2</sup>C Read command, the value written into bit 0 of the Transmit Slave Address Register (SMB I/O register, offset 04h) needs to be 0.

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

**Table 97. I<sup>2</sup>C Block Read**

Bit	Description
1	Start
8:2	Slave Address – 7 bits
9	Write
10	Acknowledge from slave
18:11	Send DATA1 register
19	Acknowledge from slave
20	Repeated Start
continued...	

Bit	Description
27:21	Slave Address – 7 bits
28	Read
29	Acknowledge from slave
37:30	Data byte 1 from slave – 8 bits
38	Acknowledge
46:39	Data byte 2 from slave – 8 bits
47	Acknowledge
–	Data bytes from slave/Acknowledge
–	Data byte N from slave – 8 bits
–	NOT Acknowledge
–	Stop

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

### **Block Write–Block Read Process Call**

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

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

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

- $M \geq 1$  byte
- $N \geq 1$  byte
- $M + N \leq 32$  bytes

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

---

#### **NOTE**

There is no STOP condition before the repeated START condition, and that a NACK signifies the end of the read transfer.

---



---

#### **NOTE**

E32B bit in the Auxiliary Control register must be set when using this protocol.

---

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

### Bus Arbitration

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

If the PCH sees that it has lost arbitration, the condition is called a collision. The PCH will set the BUS\_ERR bit in the Host Status Register, and if enabled, generate an interrupt or SMI#. The processor is responsible for restarting the transaction.

### Clock Stretching

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

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

### Bus Timeout (PCH as SMBus Master)

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

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

1. BYTE\_DONE\_STATUS bit (SMBus I/O Offset 00h, Bit 7) is set.
2. The SECOND\_TO\_STS bit (TCO I/O Offset 06h, Bit 1) is not set (this indicates that the system has not locked up).

### Interrupts/SMI#

The PCH SMBus controller uses PIRQB# as its interrupt pin. However, the system can alternatively be set up to generate SMI# instead of an interrupt, by setting the SMBUS\_SMI\_EN bit.

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

**Table 98. Enable for SMBALERT#**

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

**Table 99. Enables for SMBus Slave Write and SMBus Host Events**

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

**Table 100. Enables for the Host Notify Command**

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

### 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.

## 29.4.2 SMBus Slave Interface

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

Key features and capabilities include:

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

### NOTE

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

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

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

### Format of Slave Write Cycle

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

The following table has the values associated with the registers.

**Table 101. Slave Write Registers**

Register	Function
0	Command Register. Refer to below table for valid values written to this register.
1–3	Reserved
4	Data Message Byte 0
<i>continued...</i>	

Register	Function
5	Data Message Byte 1
6–FFh	Reserved
<i>Note:</i> The external microcontroller is responsible to make sure that it does not update the contents of the data byte registers until they have been read by the system processor. The PCH overwrites the old value with any new value received. A race condition is possible where the new value is being written to the register just at the time it is being read. The PCH will not attempt to cover this race condition (that is, unpredictable results in this case).	

**Table 102. Command Types**

Command Type	Description
0	Reserved
1	<b>WAKE/SMI#.</b> This command wakes the system if it is not already awake. If system is already awake, an SMI# is generated.
2	<b>Unconditional Powerdown.</b> This command sets the PWRBTNOR_STS bit, and has the same effect as the Powerbutton Override occurring.
3	<b>HARD RESET WITHOUT CYCLING:</b> 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	<b>HARD RESET SYSTEM.</b> 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	<b>Disable the TCO Messages.</b> 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	<b>WD RELOAD:</b> Reload watchdog timer.
7	Reserved
8	<b>SMLINK_SLV_SMI.</b> When the PCH detects this command type while in the S0 state, it sets the SMLINK_SLV_SMI_STS bit. This command should only be used if the system is in an S0 state. If the message is received during S3–S5 states, the PCH acknowledges it, but the SMLINK_SLV_SMI_STS bit does not get set. <ul style="list-style-type: none"> <li>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.</li> </ul>
9–FFh	Reserved.

### Format of Read Command

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

**Table 103. Slave Read Cycle Format**

Bit	Description	Driven By	Comment
1	Start	External Microcontroller	
2–8	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
<i>continued...</i>			

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

**Table 104. Data Values for Slave Read Registers**

Register	Bits	Description
0	7:0	Reserved
1	2:0	<b>System Power State</b> 000 = S0 011 = S3 100 = S4 101 = S5 Others = Reserved
	7:3	Reserved
2	7:0	Reserved
3	5:0	<b>Watchdog Timer current value</b> <i>Note:</i> 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	<b>Intruder Detect.</b> 1 = The Intruder Detect (INTRD_DET) bit is set. This indicates that the system cover has probably been opened.
	2:1	Reserved
	3	1 = <b>SECOND_TO_STS</b> 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	<b>SMBALERT# Status.</b> 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	<b>FWH bad bit.</b> This bit will be 1 to indicate that the FWH read returned FFh, which indicates that it is probably blank.
	1	<b>Battery Low Status.</b> 1 if the BATLOW# pin is low.

*continued...*

Register	Bits	Description
	2	<b>SYS_PWROK Failure Status:</b> This bit will be 1 if the SYSPWR_FLR bit in the GEN_PMCN_2 register is set.
	3	Reserved
	4	Reserved
	5	<b>POWER_OK_BAD:</b> Indicates the failure core power well ramp during boot/resume. This bit will be active if the SLP_S3# pin is de-asserted and PCH_PWROK pin is not asserted.
	6	<b>Thermal Trip:</b> 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" <i>Note:</i> Software should not expect a consistent value when this bit is read through SMBus/SMLink
6	7:0	Contents of the Message 1 register.
7	7:0	Contents of the Message 2 register.
8	7:0	Contents of the WDSTATUS register.
9	7:0	Seconds of the RTC
A	7:0	Minutes of the RTC
B	7:0	Hours of the RTC
C	7:0	"Day of Week" of the RTC
D	7:0	"Day of Month" of the RTC
E	7:0	Month of the RTC
F	7:0	Year of the RTC
10h–FFh	7:0	Reserved

### • Behavioral Notes

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

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

### Slave Read of RTC Time Bytes

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

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

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

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

### Format of Host Notify Command

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

### NOTE

Host software must always clear the HOST\_NOTIFY\_STS bit after completing any necessary reads of the address and data registers.

The table below shows the Host Notify format.

**Table 105. Host Notify Format**

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

### Format of Read Command

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

**Table 106. Slave Read Cycle Format**

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

**Table 107. Data Values for Slave Read Registers**

Register	Bits	Description
0	7:0	Reserved for capabilities indication. Should always return 00h. Future chips may return another value to indicate different capabilities.
1	2:0	<b>System Power State</b> 000 = S0 011 = S3 100 = S4 101 = S5 Others = Reserved
	7:3	Reserved
2	7:0	Reserved
3	5:0	<b>Watchdog Timer current value</b> <i>Note:</i> 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

*continued...*

Register	Bits	Description
4	0	<b>Intruder Detect.</b> 1 = The Intruder Detect (INTRD_DET) bit is set. This indicates that the system cover has probably been opened.
	1	<b>Temperature Event.</b> 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	<b>DOA Processor Status.</b> This bit will be 1 to indicate that the processor is dead
	3	1 = <b>SECOND_TO_STS</b> 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	<b>SMBALERT# Status.</b> 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	<b>FWH bad bit.</b> This bit will be 1 to indicate that the FWH read returned FFh, which indicates that it is probably blank.
	1	<b>Battery Low Status.</b> 1 if the BATLOW# pin is a 0.
	2	<b>SYS_PWROK Failure Status:</b> This bit will be 1 if the SYSPWR_FLR bit in the GEN_PMCON_2 register is set.
	4:3	Reserved
	5	POWER_OK_BAD. Indicates the failure core power well ramp during boot/resume. This bit will be active if the SLP_S3# pin is de-asserted and PCH_PWROK pin is not asserted.
	6	Thermal Trip. This bit will shadow the state of processor Thermal Trip status bit (CTS). Events on signal will not create a event message
	7	Reserved: Default value is "X" <i>Note:</i> Software should not expect a consistent value when this bit is read through SMBus/SMLink
6	7:0	Contents of the Message 1 register.
7	7:0	Contents of the Message 2 register.
8	7:0	Contents of the WDSTATUS register.
9	7:0	Seconds of the RTC
A	7:0	Minutes of the RTC
B	7:0	Hours of the RTC
C	7:0	"Day of Week" of the RTC
D	7:0	"Day of Month" of the RTC
E	7:0	Month of the RTC
F	7:0	Year of the RTC
10h-FFh	7:0	Reserved

**Table 108. Enables for SMBus Slave Write and SMBus Host Events**

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

## 29.5 SMBus Power Gating

SMBus shares the Power Gating Domain with LPC and Primary-to-Sideband Bridge (P2SB).

A single FET controls the single Power Gating Domain; but LPC, 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.

## 30.0 Serial Peripheral Interface (SPI)

The PCH provides two Serial Peripheral Interface (SPI) interfaces. The SPI0 interface consists of 3 chip select signals, allowing up to two flash memory devices (SPI0\_CS0# and SPI0\_CS1#) and one TPM device (SPI0\_CS2#) to be connected to the PCH. The SPI1 interface implements one Chip Select signal (SPI1\_CS#) and is intended for integrated touch implementations. The SPI0 supports either 1.8 V or 3.3 V. But SPI1 interface supports 1.8 V only.

### Acronyms

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

## 30.1 Signal Description

Name	Type	Description
<b>SPI0_CLK</b>	O	<b>SPI0 Clock:</b> SPI clock signal for the common flash/TPM interface. Supports 20 MHz, 33 MHz and 50 MHz.
<b>SPI0_CS0#</b>	O	<b>SPI0 Chip Select 0:</b> Used to select the primary SPI0 Flash device. <ul style="list-style-type: none"> <li>This signal cannot be used for any other type of device than SPI Flash.</li> </ul>
<b>SPI0_CS1#</b>	O	<b>SPI0 Chip Select 1:</b> Used to select an optional secondary SPI0 Flash device. <ul style="list-style-type: none"> <li>This signal cannot be used for any other type of device than SPI Flash.</li> </ul>
<b>SPI0_CS2#</b>	O	<b>SPI0 Chip Select 2:</b> Used to select the TPM device if it is connected to the SPI0 interface. It cannot be used for any other type of device. <ul style="list-style-type: none"> <li>TPM can be configured through Soft Straps to operate over LPC or SPI0, but no more than 1 TPM is allowed in the system.</li> </ul>
<i>continued...</i>		

Name	Type	Description
<b>SPI0_MOSI</b>	I/O	<b>SPI0 Master OUT Slave IN:</b> Defaults as a data output pin for PCH in Dual Output Fast Read mode. Can be configured with a Soft Strap as a bidirectional signal (SPI0_IO0) to support the Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes.
<b>SPI0_MISO</b>	I/O	<b>SPI0 Master IN Slave OUT:</b> Defaults as a data input pin for PCH in Dual Output Fast Read mode. Can be configured with a Soft Strap as a bidirectional signal (SPI0_IO1) to support the Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes.
<b>SPI0_IO[3:2]</b>	I/O	<b>SPI0 Data I/O:</b> 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.
GPP_D1 / <b>SPI1_CLK</b> / BK1 / SBK1	O	<b>SPI1 Clock:</b> SPI1 Clock output from PCH
GPP_D2 / <b>SPI1_MISO</b> / BK2 / SBK2	I/O	<b>SPI1 Master IN Slave OUT:</b> SPI1 serial input data from the SPI1 Touch Screen device to PCH. This Pin will also function as Output during Dual and Quad I/O operation
GPP_D3 / <b>SPI1_MOSI</b> / BK3 / SBK3	I/O	<b>SPI1 Master OUT Slave IN:</b> SPI1 serial output data from PCH to the SPI1 Touch Screen device. This Pin will also function as Input during Dual and Quad I/O operation
GPP_D21 / <b>SPI1_IO2</b>	I/O	<b>SPI1 Data I/O:</b> SPI1 I/O to comprehend the support for the Quad I/O operation
GPP_D22 / <b>SPI1_IO3</b>	I/O	<b>SPI1 Data I/O:</b> SPI1 I/O to comprehend the support for the Quad I/O operation
GPP_D0 / <b>SPI1_CS#</b> / BK0 / SBK0	O	<b>SPI1 Chip Select:</b> SPI1 chip select

## 30.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
<b>SPI0_CLK</b>	Pull-down	20 kOhm $\pm$ 30%	
<b>SPI0_MOSI</b>	Pull-up	20 kOhm $\pm$ 30%	Note
<b>SPI0_MISO</b>	Pull-up	20 kOhm $\pm$ 30%	Note
<b>SPI0_CS[2:0]#</b>	Pull-down	20 kOhm $\pm$ 30%	
<b>SPI0_IO[2:3]</b>	Pull-up	20 kOhm $\pm$ 30%	Note

### NOTE

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

### 30.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>SPI0_CLK</b>	Primary	Internal Pull-down	Driven Low	Driven Low	OFF
<b>SPI0_MOSI</b>	Primary	Hi-Z <sup>2</sup>	Internal PU, then Driven Low	Driven Low	OFF
<b>SPI0_MISO</b>	Primary	Hi-Z	Internal Pull-up	Internal Pull-up	OFF
<b>SPI0_CS0#</b>	Primary	Internal Pull-down	Driven High	Driven High	OFF
<b>SPI0_CS1#</b>	Primary	Internal Pull-down	Driven High	Driven High	OFF
<b>SPI0_CS2#</b>	Primary	Internal Pull-down	Driven High	Driven High	OFF
<b>SPI0_IO[3:2]</b>	Primary	Hi-Z (Refer Note 2)	Internal Pull-up	Internal Pull-up	OFF
Notes: 1. During reset refers to when RSMRST# is asserted. 2. SPI0_MOSI, SPI0_IO[3:2] also function as strap pins. The actual pin state during Reset is dependent on the platform Pull-up/Pull-down resistor.					

### 30.4 Functional Description

Topics Covered:

- SPI0 for Flash
- SPI0 support for TPM
- SPI1 for Touch Integration
- SPI1 support for Touch Device

#### 30.4.1 SPI0 for Flash

The Serial Peripheral Interface (SPI0) supports two SPI flash devices via 2 chip select (SPI0\_CS0# and SPI0\_CS1#). The maximum size of flash supported is determined by the SFDP-discovered addressing capability of each device. Each component can be up to 16 MB (32 MB total addressable) using 3-byte addressing. Each component can be up to 64 MB (128 MB total addressable) using 4-byte addressing. Another chip select (SPI0\_CS2#) is also available and only used for TPM on SPI support. PCH drives the SPI0 interface clock at either 20 MHz, 33 MHz, or 50 MHz and will function with SPI flash devices that support at least one of these frequencies. The SPI interface supports either 3.3 V or 1.8 V.

A SPI0 flash device supporting SFDP (Serial Flash Discovery Parameter) is required for all PCH design. A SPI0 flash device on SPI0\_CS0# with a valid descriptor must be attached directly to the PCH.

The PCH supports fast read which consist of:

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

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

The PCH SPI0 has a third chip select SPI0\_CS2# for TPM support over SPI. The TPM on SPI0 will use SPI0\_CLK, SPI0\_MISO, SPI0\_MOSI and SPI0\_CS2# SPI signals.

#### NOTE

If Boot BIOS Strap = '1' then LPC is selected as the location for BIOS. BIOS may still be placed on LPC, but all platforms with PCH require a SPI0 flash connected to SPI bus with a valid descriptor connected to Chip Select 0 in order to boot.

### 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 six separate regions.

**Table 109. SPI0 Flash Regions**

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

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

#### 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 110. Region Size Versus Erase Granularity of Flash Components**

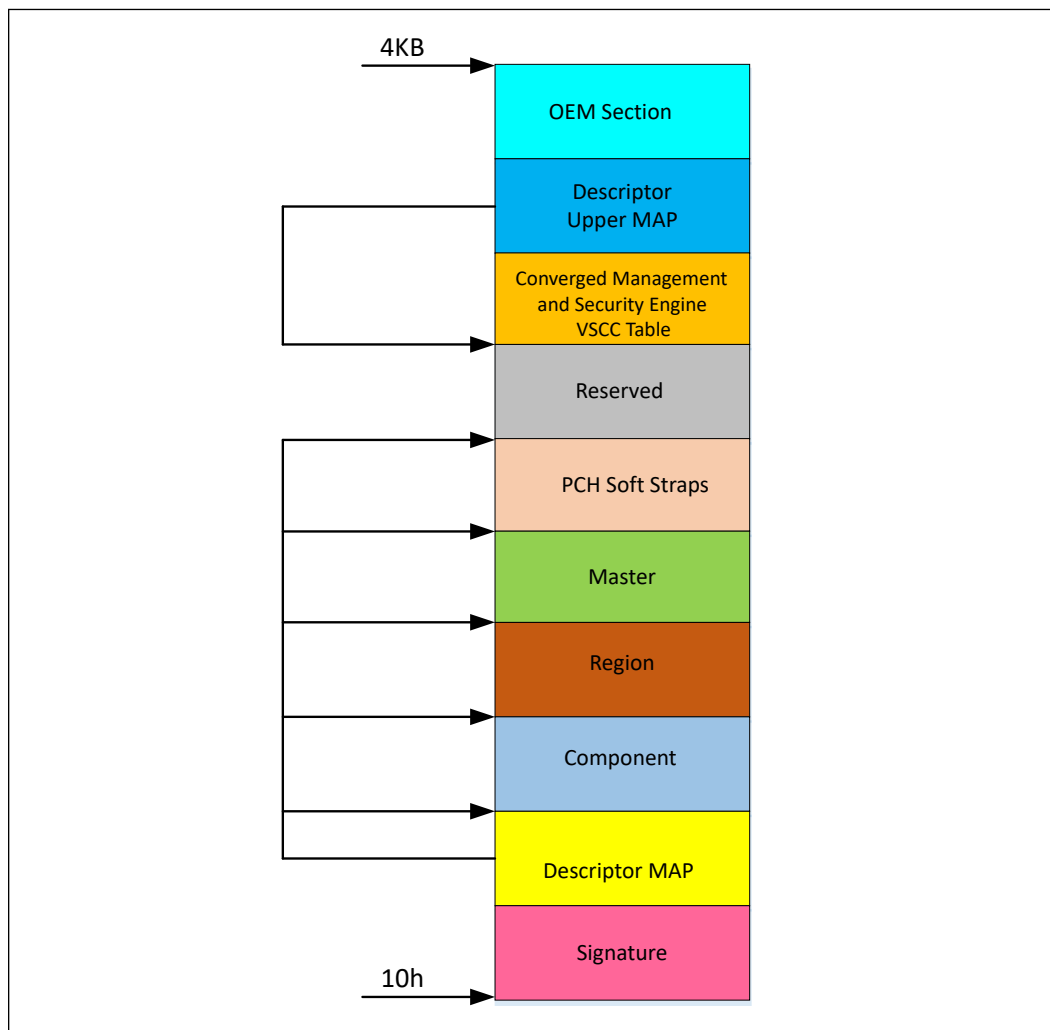
Region	Size with 4-KB Blocks	Size with 8-KB Blocks	Size with 64-KB Blocks
Descriptor	4 KB	8 KB	64 KB
GbE	8 KB	16 KB	128 KB
BIOS	Varies by Platform	Varies by Platform	Varies by Platform
Intel® CSME	Varies by Platform	Varies by Platform	Varies by Platform
EC	Varies by Platform	Varies by Platform	Varies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

**Flash Descriptor**

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

The Flash Descriptor region sections are shown in figure below.

**Figure 38. Flash Descriptor Regions**



- The Flash signature selects Descriptor Mode as well as verifies if the flash is programmed and functioning. The data at the bottom of the flash (offset 10h) must be 0FF0 A55Ah in order to be in Descriptor mode.
- The Descriptor map has pointers to the other five descriptor sections as well as the size of each.
- The component section has information about the SPI0 flash in the system including: the number of components, density of each, invalid instructions (such as chip erase), and frequencies for read, fast read and write/erase instructions.
- The Region section points to the three other regions as well as the size of each region.
- The master region contains the security settings for the flash, granting read/write permissions for each region and identifying each master by a requestor ID.
- The Reserved region between the top of the processor strap section and the bottom of the OEM Section is reserved for future chipset usages.

- The Descriptor Upper MAP determines the length and base address of the Intel® CSME VSCC Table.
- The Intel® CSME VSCC Table holds the JEDEC ID and the VSCC information of the entire SPI0 Flash supported by the NVM image.
- OEM Section is 256 bytes reserved at the top of the Flash Descriptor for use by OEM.
- **Descriptor Master Region**  
The master region defines read and write access setting for each region of the SPI0 device. The master region recognizes four masters: BIOS, Gigabit Ethernet, Intel® CSME, and EC. Each master is only allowed to do direct reads of its primary regions.

**Table 111. Region Access Control Table**

Master Access	Descriptor Region	BIOS Region	Intel® CSME	GbE Controller	EC
Intel® CSME Read access	Yes	No	Yes	Yes	No
Intel® CSME Write access	No	No	Yes	No	No
Gigabit Ethernet read access	Yes	No	No	Yes	No
Gigabit Ethernet write access	No	No	No	Yes	No
BIOS read access	Yes	Yes	No	Yes	<b>Note</b>
BIOS write access	No	Yes	No	Yes	<b>Note</b>
EC read access	Yes	Yes	No	No	Yes
EC write access	No	No	No	No	Yes

*Note:* Optional BIOS access to the EC region.

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

Region Name	Starting Address
Signature	10h
Component FCBA	30h
Regions FRBA	40h
Masters FMBA	80h
PCH Straps FPSBA	100h
MDTBA	C00h
PMC Straps	C14h
CPU Straps	C2Ch
Intel® CSME Straps	C3Ch
Register Init FIBA	340h

**Flash Access**

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

- **Direct Access**

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

**Direct Access Security**

- Requester ID of the device must match that of the primary Requester ID in the Master Section
- Calculated Flash Linear Address must fall between primary region base/limit
- Direct Write not allowed
- Direct Read Cache contents are reset to 0's on a read from a different master

- **Program Register Access**

- Program Register Accesses are not allowed to cross a 4-KB boundary and 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 masters can access the registers

### 30.4.2 SPI0 Support for TPM

The Serial Peripheral Interface (SPI0) supports a discrete TPM on the platform via its dedicated SPI0\_CS2# signal.

SPI0 controller supports accesses to SPI0 TPM at approximately 17 MHz, 30 MHz and 48 MHz depending on the PCH soft strap. 20 MHz is the reset default, a valid PCH soft strap setting overrides the requirement for the 20 MHz. SPI0 TPM device must support a clock of 20 MHz, and thus should handle 15-20 MHz. It may but is not required to support a frequency greater than 20 MHz.

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

### 30.4.3 SPI1 Support for Touch Device

The Serial Peripheral Interface (SPI1) supports SPI1 touch device via chip select (SPI1\_CS#) with Quad IO. The PCH drives the SPI1 interface clock at 30 MHz and will function with a SPI Touch device that supports this frequency.

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

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

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

### Acronyms

Acronyms	Description
GSPI	Generic Serial Peripheral Interface
LTR	Latency Tolerance Reporting

### 31.1 Signal Description

Name	Type	Description
<b>GSPI0_CS0#</b> /GPP_B15	O	<b>Generic SPI 0 Chip Select</b>
<b>GSPI0_CS1#</b> /GPP_B0	O	<b>Generic SPI 0 Chip Select</b>
<b>GSPI0_CLK</b> /GPP_B16	O	<b>Generic SPI 0 Clock</b>
<b>GSPI0_MISO</b> /GPP_B17	I	<b>Generic SPI 0 MISO</b>
<b>GSPI0_MOSI</b> /GPP_B18	O	<b>Generic SPI 0 MOSI</b> <i>Note: This signal is also utilized as a strap. Refer to <a href="#">Pin Straps</a> on page 96 for more information.</i>
<b>GSPI1_CS0#</b> /GPP_B19	O	<b>Generic SPI 1 Chip Select 0</b>
<b>GSPI1_CS1#</b> /GPP_B1/ TIME_SYNC1	O	<b>Generic SPI 1 Chip Select 1</b>
<b>GSPI1_CLK</b> /GPP_B20	O	<b>Generic SPI 1 Clock</b>
<b>GSPI1_MISO</b> /GPP_B21	I	<b>Generic SPI 1 MISO</b>
<b>GSPI1_MOSI</b> /GPP_B22	O	<b>Generic SPI 1 MOSI</b> <i>Note: This signal is also utilized as a strap. Refer to <a href="#">Pin Straps</a> on page 96 for more information.</i>
<b>GSPI2_CS0#</b> /GPP_D9 / ISH_SPI_CS#	O	<b>Generic SPI 2 Chip Select 0</b>
<b>GSPI2_CS1#</b> /GPP_D15/ ISH_UART0_RTS#	O	<b>Generic SPI 2 Chip Select 1</b>
<b>GSPI2_CLK</b> /GPP_D10 / ISH_SPI_CLK	O	<b>Generic SPI 2 Clock</b>
<b>GSPI2_MISO</b> /GPP_D11 / ISH_SPI_MISO	I	<b>Generic SPI 2 MISO</b>
<b>GSPI2_MOSI</b> /GPP_D12 / ISH_SPI_MOSI	O	<b>Generic SPI 2 MOSI</b>

## 31.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
GSPI0_MOSI	Pull Down	20 kOhm $\pm$ 30%	Internal Pull Down is not enabled by default.
GSPI1_MOSI	Pull Down	20 kOhm $\pm$ 30%	Internal Pull Down is not enabled by default.
GSPI2_MOSI	Pull Down	20 kOhm $\pm$ 30%	Internal Pull Down is not enabled by default.
GSPI0_MISO	Pull Down	20 kOhm $\pm$ 30%	
GSPI1_MISO	Pull Down	20 kOhm $\pm$ 30%	
GSPI2_MISO	Pull Down	20 kOhm $\pm$ 30%	

## 31.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
GSPI2_CS0#, GSPI2_CS1#, GSPI1_CS0#, GSPI1_CS1#, GSPI0_CS0#, GSPI0_CS1#	Primary	Undriven	Undriven	Undriven	OFF
GSPI2_CLK, GSPI1_CLK, GSPI0_CLK	Primary	Undriven	Undriven	Undriven	OFF
GSPI2_MISO, GSPI1_MISO, GSPI0_MISO	Primary	Undriven	Undriven	Undriven	OFF
GSPI2_MOSI, GSPI1_MOSI, GSPI0_MOSI	Primary	Internal Pull down	Driven Low	Internal Pull down	OFF
Note: 1. Reset reference for primary well pins is RSMRST#.					

## 31.4 Functional Description

Topics Covered:

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

### 31.4.1 Features

The GSPI interfaces support the following features:

- Full duplex synchronous serial interface
- Support Motorola's\* SPI protocol
- Operates in master mode only
- 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)

---

**NOTE**

Slave mode is not supported.

---

### 31.4.2 Controller Overview

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

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

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

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

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

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

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

### 31.4.3 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.

### 31.4.4 Reset

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

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

### 31.4.5 Power Management

#### Device Power Down Support

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

#### Latency Tolerance Reporting (LTR)

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

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

1. **Platform/HW Default Control:** This scheme is used for usage models in which the controller's state correctly informs the platform of the current latency requirements. In this scheme, the latency requirement is a function of the controller state. The latency for transmitting data to/from its connected device at a given rate while the controller is active is representative of the active latency requirements. On the other hand if the device is not transmitting or receiving data and idle, there is no expectation for end to end latency.
2. **Driver Control:** This scheme is used for usage models in which the controller state does not inform the platform correctly of the current latency requirements. If the FIFOs of the connected device are much smaller than the controller FIFOs, or the connected device's end-to-end traffic assumptions are much smaller than the latency to restore the platform from low power state, driver control should be used.

### 31.4.6 Interrupts

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

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

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

### 31.4.7 Error Handling

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

## 32.0 Testability

### 32.1 JTAG

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

#### Acronyms

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

#### References

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

#### 32.1.1 Signal Description

**Table 112. Testability Signals**

Name	Type	Description
<b>PCH_JTAG_TCK</b>	I/O	<b>Test Clock Input (TCK):</b> The test clock input provides the clock for the JTAG test logic.
<b>PCH_JTAG_TMS</b>	I/OD	<b>Test Mode Select (TMS):</b> The signal is decoded by the Test Access Port (TAP) controller to control test operations.
<b>PCH_JTAG_TDI</b>	I/OD	<b>Test Data Input (TDI):</b> Serial test instructions and data are received by the test logic at TDI.
<b>PCH_JTAG_TDO</b>	I/OD	<b>Test Data Output (TDO):</b> TDO is the serial output for test instructions and data from the test logic defined in this standard.
<i>continued...</i>		

Name	Type	Description
<b>PCH_JTAGX</b>	I/O	This pin is used to support merged debug port topologies.
<b>ITP_PMODE</b>	I/O	ITP Power Mode Indicator. This signal is used to transmit processor and PCH power/reset information to the Debugger.
<b>PREQ#</b>	I/O	<b>Probemode Request:</b> This signal is used to indicate a probe mode request send from a debugger to CPU.
<b>PRDY#</b>	I/O	<b>Probemode Ready:</b> This signal is used to acknowledge debugger that CPU in probe mode.
<b>CPU_TRST#</b>	I/O	<b>CPU Test Reset:</b> This active low signal is used to reset the test logic/Test Access Port (TAP) controller's state machine.

### 32.1.2 I/O Signal Planes and States

**Table 113. Power Planes and States for Testability Signals**

Signal Name	Power Plane	Resisters	During Reset <sup>2</sup>	Immediately after Reset <sup>2</sup>	S3/S4/S5	Deep Sx
<b>PCH_JTAG_TCK</b>	Primary	Strong Internal Pull-Down	L	L	L	OFF
<b>PCH_JTAG_TMS</b>	Primary	Internal Pull-Up	H	H	H	OFF
<b>PCH_JTAG_TDI</b>	Primary	Internal Pull-Up	H	H	H	OFF
<b>PCH_JTAG_TDO</b>	Primary	External Pull-Up	Z	Z	Z	OFF
<b>PCH_JTAGX1</b>	Primary	Internal Strong Pull-Up (as TDO i/p), Internal Strong Pull-Down (as TCK o/p)	H	H/L	H/L	OFF
<b>ITP_PMODE</b>	Primary	Internal Pull-Up	H	H	H	OFF
<b>PREQ#</b>	Primary	Internal Pull-Up	H	H	Z	OFF
<b>PRDY#</b>	Primary	Internal Pull-Up	H	H	Z	OFF
<b>CPU_TRST#</b>	Primary	Internal Pull-Up	L	L	L	OFF
Notes: 1. This signal is used in common JTAG topology to take in last device's TDO to DCI. The only planned supported topology is the Shared Topology. Thus, this pin will operate as TCK mode. 2. Reset reference for primary well pins is RSMRST#						

## 32.2 Intel® Trace Hub (Intel® TH)

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

Intel® Trace Hub is a set of silicon features with supported software API. The primary purpose is to collect trace data from different sources in the system and combine them into a single output stream with time-correlated to each other. Intel® Trace Hub

uses common hardware interface for collecting time-correlated system traces through standard destinations. Intel® Trace Hub adopts industry standard (MIPI\* STPv2) debug methodology for system debug and software development.

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

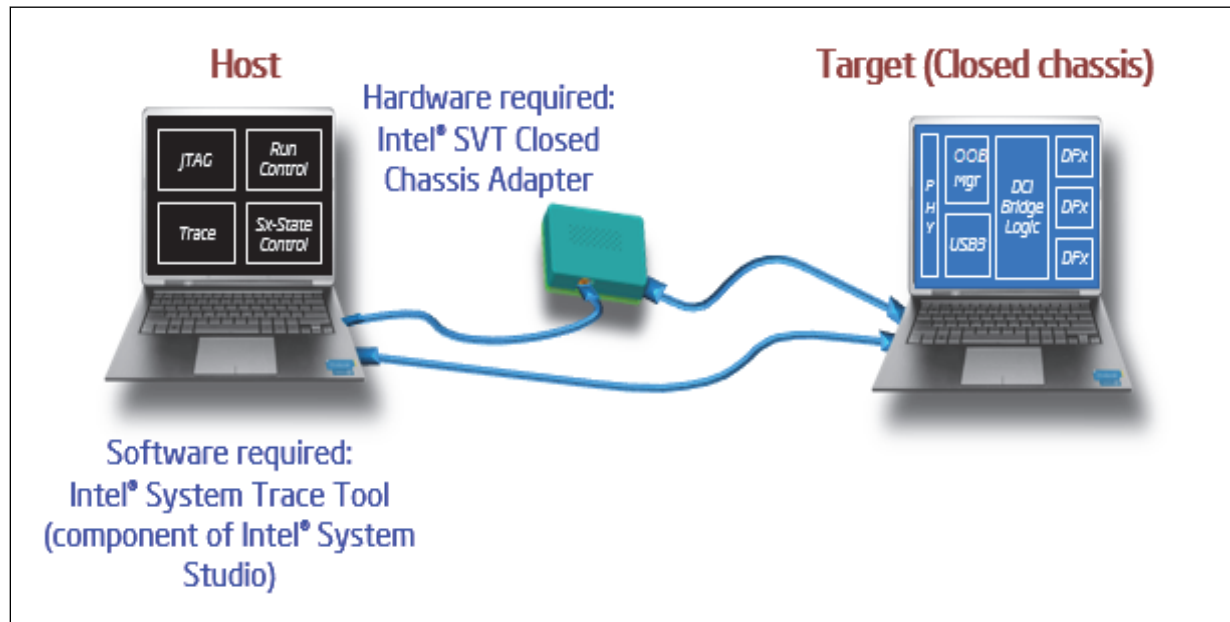
- Direct Connect Interface (DCI)
  - OOB Hosting DCI
  - USB 3.2 hosting DCI.DBC
- System Memory

There are multiple trace sources in the platform:

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

### 32.2.1 Platform Setup

**Figure 39. Platform Setup with Intel® Trace Hub**



### 32.3 Direct Connect Interface (DCI)

Direct Connect Interface (DCI) is a debug transport technology to enable closed chassis debug through any of USB 3.2 or USB 2.0 ports out from Intel silicon. Some bridging logic is embedded in the silicon to “bridge” the gap between standard I/O ports and the debug interfaces including JTAG, probe mode, hooks, trace infrastructure, and etc. To control the operation of this embedded logic, a DCI packet

based protocol is invented which controls and data can be sent or received. This protocol can operate over a few different physical transport paths to the target which known as “hosting interfaces”.

---

**NOTE**

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

---

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

- OOB Hosting DCI
- USB 3.2 or USB 2.0 Hosting DCI.DBC

Supported capabilities in DCI are:

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

Debug host software that support DCI are:

- Intel® System Studio

### 32.3.1 Out Of Band (OOB) Hosting DCI

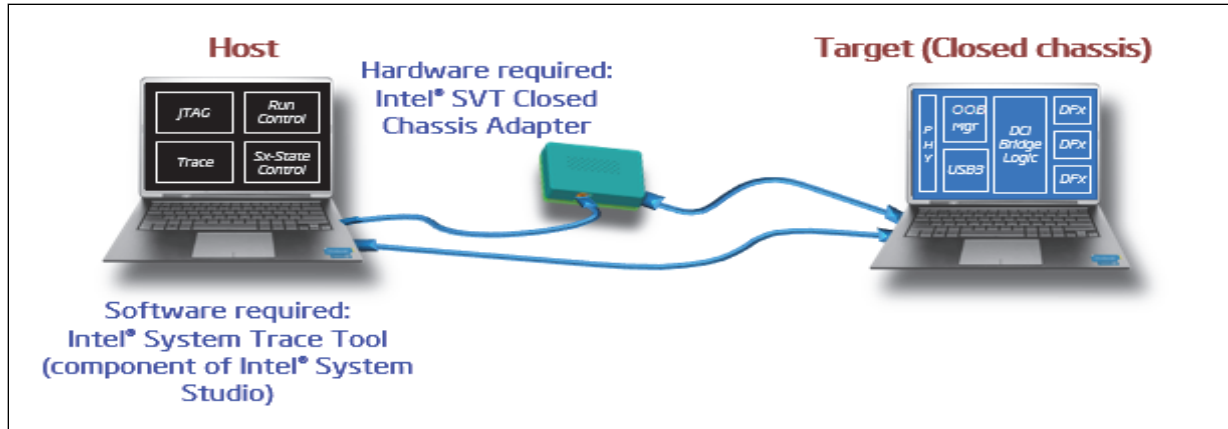
OOB was developed to provide an alternate path to convey controls and data to or from the embedded logic by connecting physically to the target through any USB 3.2 port. OOB provides an alternate side band path around the USB 3.2 Host controller, so that the embedded logic can be accessed, even when the USB controller is not alive (such as in low power states), or is malfunctioning. This path does not rely on USB protocol, link layer, or physical layer, because the xHCI functions are generally not available in such conditions. Instead, this path relies on a special adapter that developed by Intel called Intel® SVT Closed Chassis Adapter (CCA). It is a simple data transformation device. This adapter generates a OOB signaling protocol operating at up to 100 MHz and serializes data flowing through it. This adapter works together with debug host software and the embedded logic, contain a back-pressure scheme that makes both sides tolerant of overflow and starvation conditions, which is the moral equivalent of the USB link layer. This path also uses native DCI packet protocol instead of USB protocol. DCI.OOB - slower speed, CCA box needed. But survives S0ix and Sx states. Provides early boot access. Cannot tolerate re-driver circuits in its path.

### 32.3.2 USB 3.2 Gen 1x1 (5 Gb/s) and USB 2.0 Hosting DCI.DBC

It relies on Debug Class Devices (DbC) which is comprised of a set of logic that is bolted to the side of the xHCI host controller and enable the target to act the role of a USB device for debug purpose. This path uses the USB packet protocol layer, USB layer flow control and USB physical layer at 5 GHz (for USB 3.2 Gen 1x1 (5 Gb/s)) and 480MHz (for USB 2.0). DCI.DBC - fast speed. USB 3.2 Gen 2x1 (10 Gb/s) only works in S0. USB 2.0 survives S0ix and Sx states and provides early boot access.

### 32.3.3 Platform Setup

Figure 40. Platform Setup with DCI Connection



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

The PCH implements three independent UART interfaces, UART0, UART1 and UART2. Each UART interface is a 4-wire interface supporting up to 6.25 Mbit/s.

The interfaces can be used in the low-speed, full-speed, and high-speed modes. The UART communicates with serial data ports that conform to the RS-232 interface protocol.

UART2 only implements the UART Host controller and does not incorporate a DMA controller which is implemented for UART0 and UART1. Therefore, UART2 is restricted to operate in PIO mode only.

### NOTE

Bluetooth\* devices are not supported on the PCH UART interfaces.

### Acronyms

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

## 33.1 Signal Description

Name	Type	Description
<b>UART0A_RXD</b> / GPP_C8	I	<b>UART 0 Receive Data</b>
<b>UART0A_TXD</b> / GPP_C9	O	<b>UART 0 Transmit Data</b>
<b>UART0A_RTS#</b> / GPP_C10	O	<b>UART 0 Request to Send</b>
<b>UART0A_CTS#</b> / GPP_C11	I	<b>UART 0 Clear to Send</b>
<b>UART0B_RXD</b> / GPP_J5 / CNV_BRI_RSP	I	<b>Second Instant of UART 0 Receive Data</b>
<b>UART0B_TXD</b> / GPP_J6 / CNV_RGI_DT	O	<b>Second Instant of UART 0 Transmit Data</b>
<b>UART0B_RTS#</b> / GPP_J4 / CNV_BRI_DT	O	<b>Second Instant of UART 0 Request to Send</b>
<b>UART0B_CTS#</b> / GPP_J7 / CNV_RGI_RSP	I	<b>Second Instant of UART 0 Clear to Send</b>
<b>UART1_RXD</b> /ISH_UART1_RXD/ GPP_C12	I	<b>UART 1 Receive Data</b>
<b>UART1_TXD</b> /ISH_UART1_TXD/ GPP_C13	O	<b>UART 1 Transmit Data</b>
<i>continued...</i>		

Name	Type	Description
<b>UART1_RTS#</b> / ISH_UART1_RTS#/ GPP_C14	O	<b>UART 1 Request to Send</b>
<b>UART1_CTS#</b> / ISH_UART1_CTS#/ GPP_C15	I	<b>UART 1 Clear to Send</b>
<b>UART2_RXD</b> / GPP_C20	I	<b>UART 2 Receive Data</b>
<b>UART2_TXD</b> / GPP_C21	O	<b>UART 2 Transmit Data</b>
<b>UART2_RTS#</b> / GPP_C22	O	<b>UART 2 Request to Send</b>
<b>UART2_CTS#</b> / GPP_C23	I	<b>UART 2 Clear to Send</b>

## 33.2 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>UART[2:0]_RXD</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>UART[2:0]_TXD</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>UART[2:0]_RTS#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>UART[2:0]_CTS#</b>	Primary	Undriven	Undriven	Undriven	OFF

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

## 33.3 Functional Description

Topics Covered:

- Features
- UART Serial (RS-232) Protocols Overview
- 16550 8-bit Addressing - Debug Driver Compatibility
- DMA Controller
- Reset
- Power Management
- Interrupts
- Error Handling

### 33.3.1 Features

The UART interfaces support the following features:

- Up to 6.25 Mbits/s Auto Flow Control mode as specified in the 16750 standard
- Transmitter Holding Register Empty (THRE) interrupt mode
- 64-byte TX and 64-byte RX host controller FIFOs
- DMA support with 64-byte DMA FIFO per channel (up to 32-byte burst)
- Functionality based on the 16550 industry standards

- Programmable character properties, such as number of data bits per character (5-8), optional parity bit (with odd or even select) and number of stop bits (1, 1.5, or 2)
- Line break generation and detection
- DMA signaling with two programmable modes
- Prioritized interrupt identification
- Programmable FIFO enable/disable
- Programmable serial data baud rate
- Modem and status lines are independently controlled
- Programmable BAUD RATE supported (baud rate = (serial clock frequency)/(16xdivisor))

#### NOTES

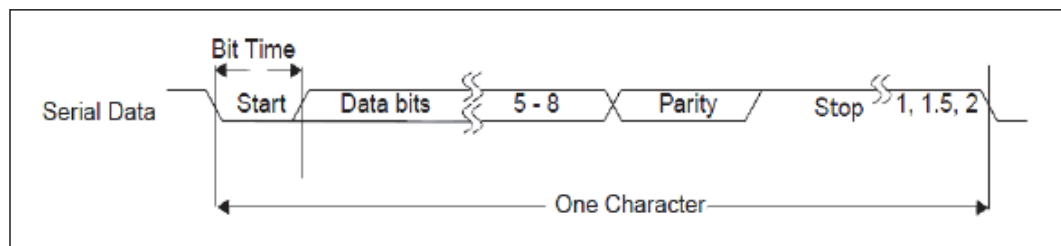
1. SIR mode is not supported.
2. External read enable signal for RAM wake up when using external RAMs is not supported.

### 33.3.2 UART Serial (RS-232) Protocols Overview

Because the serial communication between the UART host controller and the selected device is asynchronous, Start and Stop bits are used on the serial data to synchronize the two devices. The structure of serial data accompanied by Start and Stop bits is referred to as a character.

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 41. UART Serial Protocol**



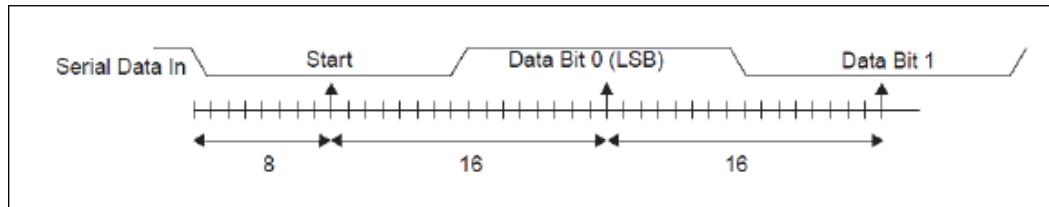
The UART Host Controller Line Control Register (LCR) is used to control the serial character characteristics. The individual bits of the data word are sent after the Start bit, starting with the least significant bit (LSB). These are followed by the optional parity bit, followed by the Stop bit(s), which can be 1, 1.5, or 2.

The Stop bit duration implemented by UART host controller may appear longer due to idle time inserted between characters for some configurations and baud clock divisor values in the transmit direction.

All bit in the transmission (with exception to the half stop bit when 1.5 stop bits are used) are transmitted for exactly the same time duration (which is referred to as Bit Period or Bit Time). One Bit Time equals to 16 baud clocks.

To ensure stability on the line, the receiver samples the serial input data at approximately the midpoint of the Bit Time once the start bit has been detected.

**Figure 42. UART Receiver Serial Data Sample Points**



### 33.3.3 16550 8-bit Addressing - Debug Driver Compatibility

The PCH UART host controller is not compatible with legacy UART 16550 debug-port drivers. The UART host controller operates in 32-bit addressing mode only. UART 16550 legacy drivers only operate with 8-bit (byte) addressing. In order to provide compatibility with standard in-box legacy UART drivers a 16550 Legacy Driver mode has been implemented in the UART controller that will convert 8-bit addressed accesses from the 16550 legacy driver to the 32-bit addressing that the UART host controller supports.

#### NOTE

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

### 33.3.4 DMA Controller

The UART controllers 0 and 1 (UART0 and UART1) have an integrated DMA controller. Each channel contains a 64-byte FIFO. Max. burst size supported is 32 bytes.

UART controller 2 (UART2) only implements the host controllers and does not incorporate a DMA. Therefore, UART2 is restricted to operate in PIO mode only.

#### DMA Transfer and Setup Modes

The DMA can operate in the following modes:

1. **Memory to Peripheral Transfers:** This mode requires that the peripheral control the flow of the data to itself.
2. **Peripheral to Memory Transfer:** This mode requires that the peripheral control the flow of the data from itself.

The DMA supports the following modes for programming:

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

### Channel Control

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

### 33.3.5 Reset

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

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

### 33.3.6 Power Management

#### Device Power Down Support

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

#### Latency Tolerance Reporting (LTR)

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

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

1. **Platform/HW Default Control:** This scheme is used for usage models in which the controller's state correctly informs the platform of the current latency requirements. In this scheme, the latency requirement is a function of the

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

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

### 33.3.7 Interrupts

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

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

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

### 33.3.8 Error Handling

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

## 34.0 Universal Serial Bus (USB)

The PCH implements an USB 3.2 xHCI 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).

### NOTE

Each walk-up USB 3.2 capable port must have USB 3.2 signaling and USB 2.0 signaling.

### Acronyms

Acronyms	Description
xHCI	eXtensible Host Controller Interface

### References

Specification	Location
USB 3.2 Specification	www.usb.org
USB 3.1 Specification	
USB 3.0 Specification	
USB 2.0 Specification	

## 34.1 Signal Description

Name	Type	Description
<b>USB31_1_RXN/ USB31_1_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 1:</b> 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.
<b>USB31_1_TXN/ USB31_1_TXP</b>	O	<b>USB 3.2 Differential Transmit Pair 1:</b> 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. 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.
<b>USB31_2_RXN/ USB31_2_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 2:</b> 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.
<b>USB31_2_TXN/ USB31_2_TXP</b>	O	<b>USB 3.2 Differential Transmit Pair 2:</b> 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.
<i>continued...</i>		

Name	Type	Description
<b>USB31_3_RXN/ USB31_3_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 3:</b> 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.
<b>USB31_3_TXN/ USB31_3_TXP</b>	O	<b>USB 3.2 Differential Transmit Pair 3:</b> 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.
<b>USB31_4_RXN/ USB31_4_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 4:</b> 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.
<b>USB31_4_TXN/ USB31_4_TXP</b>	O	<b>USB 3.2 Differential Transmit Pair 4:</b> 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.
<b>USB31_5_RXN USB31_5_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 5:</b> 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.
<b>USB31_5_TXN USB31_5_TXP</b>	O	<b>USB 3.2 Differential Transmit Pair 5:</b> 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.
<b>USB31_6_RXN USB31_6_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 6:</b> 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.
<b>USB31_6_TXN USB31_6_TXP</b>	O	<b>USB 3.2 Differential Transmit Pair 6:</b> 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.
<b>USB31_7_RXN / PCIE1_RXN, USB31_7_RXP / PCIE1_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 7:</b> 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.
<b>USB31_7_TXN / PCIE1_TXN, USB31_7_TXP / PCIE1_TXP</b>	O	<b>USB 3.2 Differential Transmit Pair 7:</b> 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.
<b>USB31_8_RXN / PCIE2_RXN, USB31_8_RXP / PCIE2_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 8:</b> 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.
<b>USB31_8_TXN / PCIE2_TXN, USB31_8_TXP / PCIE2_TXP</b>	O	<b>USB 3.2 Differential Transmit Pair 8:</b> 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.
<b>USB31_9_RXN / PCIE3_RXN, USB31_9_RXP / PCIE3_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 9:</b> 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.
<b>USB31_9_TXN / PCIE3_TXN, USB31_9_TXP / PCIE3_TXP</b>	O	<b>USB 3.2 Differential Transmit Pair 9:</b> 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.
<b>USB31_10_RXN / PCIE4_RXN, USB31_10_RXP / PCIE4_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 10:</b> 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.
<i>continued...</i>		

Name	Type	Description
<b>USB31_10_TXN /</b> PCIE4_TXN, <b>USB31_10_TXP /</b> PCIE4_TXP	O	<b>USB 3.2 Differential Transmit Pair 10:</b> 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.
<b>USB2P_1/</b> <b>USB2N_1</b>	I/O	<b>USB 2.0 Port 1 Transmit/Receive Differential Pair 1:</b> 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.
<b>USB2P_2/</b> <b>USB2N_2</b>	I/O	<b>USB 2.0 Port 2 Transmit/Receive Differential Pair 2:</b> 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.
<b>USB2P_3/</b> <b>USB2N_3</b>	I/O	<b>USB 2.0 Port 3 Transmit/Receive Differential Pair 3:</b> 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.
<b>USB2P_4/</b> <b>USB2N_4</b>	I/O	<b>USB 2.0 Port 4 Transmit/Receive Differential Pair 4:</b> 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.
<b>USB2P_5/</b> <b>USB2N_5</b>	I/O	<b>USB 2.0 Port 5 Transmit/Receive Differential Pair 5:</b> 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.
<b>USB2P_6/</b> <b>USB2N_6</b>	I/O	<b>USB 2.0 Port 6 Transmit/Receive Differential Pair 6:</b> 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.
<b>USB2P_7/</b> <b>USB2N_7</b>	I/O	<b>USB 2.0 Port 7 Transmit/Receive Differential Pair 7:</b> 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.
<b>USB2P_8/</b> <b>USB2N_8</b>	I/O	<b>USB 2.0 Port 8 Transmit/Receive Differential Pair 8:</b> 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.
<b>USB2P_9/</b> <b>USB2N_9</b>	I/O	<b>USB 2.0 Port 9 Transmit/Receive Differential Pair 9:</b> 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.
<b>USB2P_10/</b> <b>USB2N_10</b>	I/O	<b>USB 2.0 Port 10 Transmit/Receive Differential Pair 10:</b> 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.
<b>USB2P_11/</b> <b>USB2N_11</b>	I/O	<b>USB 2.0 Port 11 Transmit/Receive Differential Pair 11:</b> 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.
<b>USB2P_12/</b> <b>USB2N_12</b>	I/O	<b>USB 2.0 Port 12 Transmit/Receive Differential Pair 12:</b> 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.
<b>USB2P_13/</b> <b>USB2N_13</b>	I/O	<b>USB 2.0 Port 13 Transmit/Receive Differential Pair 13:</b> 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.
<b>USB2P_14/</b> <b>USB2N_14</b>	I/O	<b>USB 2.0 Port 14 Transmit/Receive Differential Pair 14:</b> 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.
<b>USB2_OC0# /</b> GPP_E9	I	<b>Overcurrent Indicators:</b> This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred. When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required.
continued...		

Name	Type	Description
		<i>Notes:</i> 1. OC# pins are not 5V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..
<b>USB2_OC1#</b> / GPP_E10	I	<b>Overcurrent Indicators:</b> This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred. When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required. <i>Notes:</i> 1. OC# pins are not 5V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..
<b>USB2_OC2#</b> / GPP_E11	I	<b>Overcurrent Indicators:</b> This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred. When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required. <i>Notes:</i> 1. OC# pins are not 5V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..
<b>USB2_OC3#</b> / GPP_E12	I	<b>Overcurrent Indicators:</b> This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred. When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required. <i>Notes:</i> 1. OC# pins are not 5V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..
<b>USB2_OC4#</b> / GPP_F15	I	<b>Overcurrent Indicators:</b> This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred. When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required. <i>Notes:</i> 1. OC# pins are not 5V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..
<b>USB2_OC5#</b> / GPP_F16	I	<b>Overcurrent Indicators:</b> This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred. When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required. <i>Notes:</i> 1. OC# pins are not 5V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..
<b>USB2_OC6#</b> / GPP_F17	I	<b>Overcurrent Indicators:</b> This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred. When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required.
<b>continued...</b>		

Name	Type	Description
		<i>Notes:</i> 1. OC# pins are not 5V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..
<b>USB2_OC7# / GPP_F18</b>	I	<b>Overcurrent Indicators:</b> 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 k $\Omega$ 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. <i>Notes:</i> 1. OC# pins are not 5V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..
<b>USB2_VBUSSENSE</b>	I	VBUS Sense for USB Device mode. <i>Note:</i> This HW signal is not used on the PCH for USB device mode functionality. This signal should be connected to ground.
<b>USB2_ID</b>	I	ID detect for USB Device mode. <i>Note:</i> This HW signal is not used on the PCH for dual role mode selection. The switching of USB port role is done by EC through the eSPI Out of Band (OOB) message or LPC SCI/SMI to notify BIOS to set the SPR registers in the PCH. This signal should be connected to ground.
<b>USB2_COMP</b>	I	USB Resistor Bias, analog connection points for an external resistor 113 $\Omega$ $\pm$ 1% connected to GND.

**NOTE**

It is not recommended to route the USB 3.2 signals to the USB connector for USB ports that is USB 2.0 capable only.

## 34.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
<b>USB2N_[140:1]</b>	Internal Pull-down	14.25–24.8 kohm	1
<b>USB2P_[140:1]</b>	Internal Pull-down	14.25–24.8 kohm	1
<b>USB2_ID</b>	Internal Weak Pull-up	14.25–24.8 kohm	If this signal is not in use, then the pin shall be connected directly to ground.

*Note:* 1. Series resistors (45 ohm  $\pm$ 10%).

## 34.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>2</sup>	Immediately after Reset <sup>2</sup>	S3/S4/S5	Deep Sx
<b>USB31_[10:1]_RXN</b> <b>USB31_[10:1]_RXP</b>	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
<b>USB31_[10:1]_TXN</b> <b>USB31_[10:1]_TXP</b>	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF

*continued...*

Signal Name	Power Plane	During Reset <sup>2</sup>	Immediately after Reset <sup>2</sup>	S3/S4/S5	Deep Sx
<b>USB2N_[14:1]</b>	DSW	Internal Pull-down	Internal Pull-down	Internal Pull-down	Internal Pull-down
<b>USB2P_[14:1]</b>	DSW	Internal Pull-down	Internal Pull-down	Internal Pull-down	Internal Pull-down
<b>USB2_OC0#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>USB2_OC1#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>USB2_OC2#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>USB2_OC3#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>USB2_OC4#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>USB2_OC5#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>USB2_OC6#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>USB2_OC7#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>USB2_VBUSSENSE</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>USB2_ID<sup>1</sup></b>	Primary	Internal Pull-UP	Undriven/Internal Pull-UP	Undriven/Internal Pull-UP	OFF
<b>USB2_COMP</b>	Primary	Undriven	Undriven	Undriven	OFF

Notes: 1. The USB2\_ID pin is pulled up internally.  
2. Reset reference for primary well pins is RSMRST#, DSW well pins is DSW\_PWROK, and RTC well pins is RTCRST#.

## 34.4 Functional Description

Topics Covered:

- eXtensible Host Controller Interface (xHCI) Controller (D20:F0)
- USB Dual Role Support - eXtensible Device Controller Interface (xHCI) Controller (D20:F1)
- Supported USB 2.0 Ports

### 34.4.1 eXtensible Host Controller Interface (xHCI) Controller (D20:F0)

The PCH contains an eXtensible Host Controller Interface (xHCI) which supports up to 14 USB 2.0 ports and up to 10 USB 3.2 ports with board routing, ACPI table and BIOS considerations are required. This controller allows data transfers of up to 10 Gb/s for USB 3.2 Gen 2x1 ports and 5 Gb/s for USB 3.2 Gen 1x1 ports. The controller supports SuperSpeed USB 10 Gbps, SuperSpeed USB 5 Gbps, High-Speed (HS), Full-Speed (FS) and Low-Speed (LS) traffic on the bus. The xHCI supports USB Debug port on all USB 3.2 capable ports. The xHCI also supports USB Attached SCSI Protocol (UASP). The PCH also supports dual role capability.

### 34.4.2 USB Dual Role Support - eXtensible Device Controller Interface (xHCI) Controller (D20:F1)

The PCH also supports Dual Role Capability. The xHCI can now be paired with a standalone eXtensible Device Controller Interface (xDCI) to provide dual role functionality. The USB subsystem incorporates a USB 3.2 Gen 1x1 (5 Gb/s) xDCI controller. The Device controller support SuperSpeed USB 5 Gbps, High-Speed (HS). This controller is instantiated as a separate PCI function. The PCH USB implementation is compliant to the Device specification and supports host/device through USB Type-C\* connector.

The xDCI shares all USB ports with the host controller, with 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.

### 34.4.3 Supported USB 2.0 Ports

Due to the USB 2.0 port requirement for integrated Bluetooth\* functionality with the integrated Intel® Wireless-AC (CNVi) solution, PCH-H USB 2.0 port 14<sup>1</sup> will be available for the following:

- Supported Port
  - USB 2.0 port 14 will be enabled

The total USB 2.0 port availability for a given SKU will also take into account the USB 2.0 port requirement for integrated Bluetooth\* functionality. The following table describes the number of port supported and the associated port number enabled per SKU.

**Figure 43. Supported USB 2.0 Ports**

CHIPSET SKU	Max USB 2.0 Nbr of Ports	USB 2.0 P1	USB 2.0 P2	USB 2.0 P3	USB 2.0 P4	USB 2.0 P5	USB 2.0 P6	USB 2.0 P7	USB 2.0 P8	USB 2.0 P9	USB 2.0 P10	USB 2.0 P11	USB 2.0 P12	USB 2.0 P13	USB 2.0 P14 (or Intel® Wireless-AC)	USB r1	USB r2
Z490	14																
Q470	14																
H470	14																
W480	14																
HM470	14																
QM480	14																
WM490	14																

Port Disabled
  Port Enabled

## 35.0 Connectivity Integrated (CNVi)

Connectivity Integrated (CNVi) is a general term referring to a family of connectivity solutions which are based on the Connectivity Controller family. The common component of all these solutions is the Connectivity Controller IP.

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

- The containing chip (PCH which contains the Connectivity Controller IP)
- Buttress (as applicable to each platform)
- Companion RF chip that is in a pre-certified module (that is, M.2) or soldered as chip on board.

### Acronyms

Acronyms	Description
CNVi	Connectivity Integrated
IP	Literally, Intellectual Property. IP refers to architecture, design, validation, and software components collectively delivered to enable one or more specific features
MFUART	Multifunction Universal Asynchronous Receiver/Transmitter
RGI	Radio Generic interface
UART	Universal Asynchronous Receiver/Transmitter

### References

Specification	Location
M.2 Specification	<a href="https://pcisig.com/specifications/pciexpress/M.2_Specification/">https://pcisig.com/specifications/pciexpress/M.2_Specification/</a>
MIPI® Alliance specification for D-PHY v1.1	<a href="http://www.mipi.org/specifications/">http://www.mipi.org/specifications/</a>

## 35.1 Signal Description

Name	Type	Description
<b>GPIO Fixed Function</b>		
GPP_D5 / I2S2_SFRM / <b>CNV_RF_RESET#</b>	I/O	For CNVi: RF companion (CRF) reset signal, active low, must be low and glitch free at power up. For standard CNV with UART host support: Optional Bluetooth I <sup>2</sup> S bus sync.
GPP_D6 / I2S2_TXD / <b>MODEM_CLKREQ</b>	O	For CNVi: Clock request signal. Used to request the RF companion clock (38.4M Ref clock) for Pulsar and the PCH. For standard CNV with UART host Bluetooth support: Optional Bluetooth I <sup>2</sup> S bus data output (input to Bluetooth* module).
GPP_D7 / <b>I2S2_RXD</b>	I	For CNVi: Unused. For standard CNV with UART host support: Optional Bluetooth I <sup>2</sup> S bus data output (input to Bluetooth module).
<i>continued...</i>		

Name	Type	Description
GPP_D8 / <b>I2S2_SCLK</b>	I	For CNVi: Unused. For standard CNV with UART host support: Optional Bluetooth I <sup>2</sup> S bus clock (input to Bluetooth module).
GPP_J0 / <b>CNV_PA_BLANKING</b>	I	For CNVi and standard CNV: Optional WLAN/Bluetooth-WWAN coexistence signal COEX3. Co-existence signal for external GNSS solution
GPP_J8 / <b>CNV_MFUART2_RXD</b>	I	For CNVi and standard CNV: Optional WLAN/Bluetooth-WWAN coexistence signal COEX2.
GPP_J9 / <b>CNV_MFUART2_TXD</b>	O	For CNVi and standard CNV: Optional WLAN/Bluetooth-WWAN coexistence signal COEX1.
GPP_J4 / <b>CNV_BRI_DT</b>	O	For CNVi: BRI bus TX. For standard CNV CNV with UART host support: Bluetooth UART RTS#
GPP_J5 / <b>CNV_BRI_RSP</b>	I	For CNVi: BRI bus RX. For standard CNV CNV with UART host support: Bluetooth UART RXD
GPP_J6 / <b>CNV_RGI_DT</b>	O	For CNVi: RGI bus TX. For standard CNV with UART host support: Bluetooth UART TXD
GPP_J7 / <b>CNV_RGI_RSP</b>	I	For CNVi: RGI bus RX. For standard CNV with UART host support: Bluetooth UART CTS#
<b>GPIO Selectable Function</b>		
<b>GPP_C2</b>	O	UART_BT_WAKE output. Optional to connect to Bluetooth Wake on Bluetooth Module. This is the recommended GPIO but other GPIOs can be selected for this function.
<b>GPP_B3</b>	O	BT_KILL output. Optional to connect to a Bluetooth KILL pin on the Bluetooth module. This is the recommended GPIO but other GPIOs can be selected for this function.
<b>GPP_C5</b>	O	For CNVi: Unused Wi-Fi PCIe* host wake output for standard CNV. Optional to connect to a WLAN PCIe* host wake pin on the Wi-Fi module. This is the recommended GPIO but other GPIOs can be selected for this function.
<b>GPP_B4</b>	O	WLAN_KILL output. Optional to connect to a WLAN KILL pin on the Wi-Fi module. This is the recommended GPIO but other GPIOs can be selected for this function.
<b>GPP_B8</b>	O	For CNVi: Unused Wi-Fi PCIe* clock request output for standard CNV. Optional to connect to a WLAN PCIe* clock request pin on the Wi-Fi module. This is the recommended GPIO but other GPIOs can be selected for this function
<b>Fixed Special Purpose I/O</b>		
<b>CNV_WT_CLKP</b>	O	CNVio bus TX CLK+
<b>CNV_WT_CLKN</b>	O	CNVio bus TX CLK-
<b>CNV_WT_D0P</b>	O	CNVio bus Lane 0 TX+
<b>CNV_WT_D0N</b>	O	CNVio bus Lane 0 TX-
<b>CNV_WT_D1P</b>	O	CNVio bus Lane 1 TX+
<b>CNV_WT_D1N</b>	O	CNVio bus Lane 1 TX-
<b>CNV_WR_CLKP</b>	I	CNVio bus RX CLK+
<b>CNV_WR_CLKN</b>	I	CNVio bus RX CLK-
<b>continued...</b>		

Name	Type	Description
<b>CNV_WR_D0P</b>	I	CNVio bus Lane 0 RX+
<b>CNV_WR_D0N</b>	I	CNVio bus Lane 0 RX-
<b>CNV_WR_D1P</b>	I	CNVio bus Lane 1 RX+
<b>CNV_WR_D1N</b>	I	CNVio bus Lane 1 RX-
<b>CLKIN_XTAL</b>	I	Single-ended clock signal input for CNVi (38.4 MHz). Should be connected to the reference clock output of the RF companion (CRF)
<b>CNV_WT_RCOMP</b>	I	An analog pin, direct bump for external compensation resistor (RCOMP) for CNVio Interface. Connected to supply VCCDPHY_1P24
<b>Selectable Special Purpose I/O</b>		
<b>USB2P_14</b>		Bluetooth USB host bus (positive) for standard CNV. Optional to connect to a Bluetooth USB+ pin on the Bluetooth module. Port 14 is the recommended port but other USB2 ports can be selected for this function.
<b>USB2N_14</b>		Bluetooth USB host bus (negative) for standard CNV. Optional to connect to a Bluetooth USB+ pin on the Bluetooth module. Port 14 is the recommended port but other USB2 ports can be selected for this function.
<b>PCIE8_TXP</b>	O	Wi-Fi PCIe* host bus TX (positive) for standard CNV. Optional to connect to a Wi-Fi PCIe* PERp0 pin on the Wi-Fi module. This is the recommended port but other PCIe* ports can be selected for this function.
<b>PCIE8_TXN</b>	O	Wi-Fi PCIe* host bus TX (negative) for standard CNV. Optional to connect to a Wi-Fi PCIe* PERn0 pin on the Wi-Fi module. This is the recommended port but other PCIe* ports can be selected for this function.
<b>PCIE8_RXP</b>	I	Wi-Fi PCIe* host bus RX (positive) for standard CNV. Optional to connect to a Wi-Fi PCIe* PETp0 pin on the Wi-Fi module. This is the recommended port but other PCIe* ports can be selected for this function.
<b>PCIE8_RXN</b>	I	Wi-Fi PCIe* host bus RX (negative) for standard CNV. Optional to connect to a Wi-Fi PCIe* PETn0 pin on the Wi-Fi module. This is the recommended port but other PCIe* ports can be selected for this function.
<b>CLKOUT_PCIE_P3</b>	O	Wi-Fi PCIe* host bus clock (positive) for standard CNV. Optional to connect to a Wi-Fi PCIe* REFCLKp pin on the Wi-Fi module. This is the recommended clock signal but other PCIe* clocks can be selected for this function.
<b>CLKOUT_PCIE_N3</b>	O	Wi-Fi PCIe* host bus clock (negative) for standard CNV. Optional to connect to a Wi-Fi PCIe* REFCLKp pin on the Wi-Fi module. This is the recommended clock signal but other PCIe* clocks can be selected for this function.
<b>CL_RST#</b>	O	Wi-Fi Controller Link host bus reset for standard CNV with Controller Link support (Intel® vPro™). Optional to connect to a Wi-Fi CLINK reset pin on the Intel® vPro™ Wi-Fi module.
<b>CL_DATA</b>	I/O	Wi-Fi Controller Link host bus data for standard CNV with Controller Link support (Intel® vPro™). Optional to connect to a Wi-Fi CLINK data pin on the Intel® vPro™ Wi-Fi module.
<b>CL_CLK</b>	O	Wi-Fi Controller Link host bus clock for standard CNV with Controller Link support (Intel® vPro™). Optional to connect to a Wi-Fi CLINK clock pin on the Intel® vPro™ Wi-Fi module.

## 35.2 Platform PU/PD requirements

The OEM should apply pull-ups and pull-downs on the platform side according to the below table

I/F	Signals	PU/PD in Platform	Comments
BRI/RGI (Bluetooth* UART)	RGI_DT	PU (20 kOhm)	This pull is required so that the SOC will be able to reliably detect that the CRF is present at power-up. However, it is possible to increase the resistor to 50K or even to 100K instead of 20K.
Init signals	CNV_RF_RESET#	PU (75 kOhm)	It is highly encouraged to increase this resistor (or allow to switch it off when CNVi is active; not sure this is possible at the platform level). This resistor consumes power (43uW) all the time.
38.4 Ref clock	RefCLK	PD (10 kOhm) (for Jefferson Peak)	If used (a platform-level decision); not supported and not connected by Harrison Peak.
A4WP indication	A4WP_PRESENT	PD (75 kOhm)	Native function A4WP is not supported. The pin can instead be used as GPIO (when BIOS programs the pin to GPIO functionality). It is recommended to have an external pull down (75 kOhm) on the pin regardless of the pin being used or not to minimize power consumption. If the pin is used as GPIO, there should NOT be any on-board device driving the pin high until BIOS programs it to GPIO functionality

The OEM must avoid using a specific PU/PD when not needed or when required to not be used. Unless this rule is followed, a back-bias condition will result, where the IO is getting voltage before the device side is ready for it.

## 35.3 I/O Signal Planes and States

Signal Name	Power plane	During Reset <sup>1</sup>	Immediately After Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>I2S2_RXD</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>I2S2_SCLK</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>I2S2_TXD</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>I2S2_SFRM</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>CNV_RF_RESET#</b>	Primary	Driven	Driven	Driven	OFF
<b>MODEM_CLKREQ</b>	Primary	Driven	Driven	Driven	OFF
<b>CNV_PA_BLANKING</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>CNV_MFUART2_RXD</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>CNV_MFUART2_TXD</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>CNV_BRI_DT</b>	Primary	Driven	Driven	Driven	OFF
<b>CNV_BRI_RSP</b>	Primary	Powered (input)	Powered (input)	Powered (input)	OFF
<b>continued...</b>					

Signal Name	Power plane	During Reset <sup>1</sup>	Immediately After Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>CNV_RGI_DT</b>	Primary	Driven	Driven	Driven	OFF
<b>CNV_RGI_RSP</b>	Primary	Powered (input)	Powered (input)	Powered (input)	OFF
<b>CNV_WT_CLKP</b>	Primary	Undriven	Undriven	Driven	OFF
<b>CNV_WT_CLKN</b>	Primary	Undriven	Undriven	Driven	OFF
<b>CNV_WT_D0P</b>	Primary	Undriven	Undriven	Driven	OFF
<b>CNV_WT_D0N</b>	Primary	Undriven	Undriven	Driven	OFF
<b>CNV_WT_D1P</b>	Primary	Undriven	Undriven	Driven	OFF
<b>CNV_WT_D1N</b>	Primary	Undriven	Undriven	Driven	OFF
<b>CNV_WR_CLKP</b>	Primary	Undriven	Undriven	Powered (input)	OFF
<b>CNV_WR_CLKN</b>	Primary	Undriven	Undriven	Powered (input)	OFF
<b>CNV_WR_D0P</b>	Primary	Undriven	Undriven	Powered (input)	OFF
<b>CNV_WR_D0N</b>	Primary	Undriven	Undriven	Powered (input)	OFF
<b>CNV_WR_D1P</b>	Primary	Undriven	Undriven	Powered (input)	OFF
<b>CNV_WR_D1N</b>	Primary	Undriven	Undriven	Powered (input)	OFF
<b>CNV_WT_RCOMP</b>	Primary	Undriven	Undriven	Driven	OFF
<b>CLKIN_XTAL_LCP</b>	Primary	Depends on MODEM_CLKR EQ	Depends on MODEM_CLKREQ	Depends on MODEM_CLKR EQ	OFF

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

## 35.4 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\*).

Connectivity Controller core includes only the digital part of the connectivity function, while the CRF cores handle some digital and most of 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.

## 36.0 Secure Digital eXtended Capacity (SDXC)

The SDXC controller is to connect to an external detachable storage devices. It supports SDXC specification version 3.01.

### Acronyms

Acronyms	Description
SDXC	Secure Digital eXtended Capacity

### References

Specification	Location
SDXC Specifications	<a href="http://www.sdcard.org">http://www.sdcard.org</a>

## 36.1 Signal Description

Name	Type	Description	Voltage
<b>SD_CMD</b> /GPP_G0	I/O	SD Command/Response	3.3 V or 1.8 V
<b>SD_DATA0</b> /GPP_G1	I/O	SD Data	3.3 V or 1.8 V
<b>SD_DATA1</b> /GPP_G2	I/O	SD Data	3.3 V or 1.8 V
<b>SD_DATA2</b> /GPP_G3	I/O	SD Data	3.3 V or 1.8 V
<b>SD_DATA3</b> /GPP_G4	I/O	SD Data	3.3 V or 1.8 V
<b>SD_CD#</b> /GPP_G5	I	SDXC Detect	3.3 V or 1.8 V
<b>SD_CLK</b> /GPP_G6	O	SD clock	3.3 V or 1.8 V
<b>SD_WP</b> /GPP_G7	I	SDXC write protect	3.3 V or 1.8 V
<b>SD_VDD1_PWR_EN#</b> / ISH_GP7 /GPP_A17	O	SDXC power enable for 3.3 V <i>Note:</i> Signal assertion logic can be changed by BIOS. (Active High is default assertion logic)	1.8 V
<b>SD_3P3_RCOMP</b>	I/O	Impedance compensation for 3.3 V operation of SD card and GPIO buffers. External reference resistor (200 Ohm +/- 1% pull down to ground) is required, regardless of SD card interface is used or not.	NA
<b>SD_1P8_RCOMP</b>	I/O	Impedance compensation for 1.8 V operation of SD card and GPIO buffers. External reference resistor (200 Ohm +/- 1% pull down to ground) is required, regardless of SD card interface is used or not. The RCOMP resistor can be shared with SD_3P3_RCOMP.	NA

## 36.2 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
SD_CMD	Primary	Undriven	Undriven	Undriven	OFF
SD_DATA[3:0]	Primary	Undriven	Undriven	Undriven	OFF
SD_CD#	Primary	Undriven	Undriven	Undriven	OFF
SD_CLK	Primary	Undriven	Undriven	Undriven	OFF
SD_WP	Primary	Undriven	Undriven	Undriven	OFF
SD_VDD1_PWR_EN#	Primary	Undriven	Driven	Undriven	OFF
SD_3P3_RCOMP	Primary	Undriven	Undriven	Undriven	OFF
SD_1P8_RCOMP	Primary	Undriven	Undriven	Undriven	OFF

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

## 36.3 Functional Description

The SDXC controller handles SD Protocol at transmission, packing data, adding cyclic redundancy check (CRC), start/end bit, and checking for transaction format correctness. The SDXC main use case is to connect to an external detachable storage device. It supports SDXC card specification version 3.01. Both 1.8V and 3.3V signaling is supported. Additional information can be obtained from the SDXC 3.01 specification.

The following chart maps the working modes of SDXC.

**Table 114. SD Working Modes**

SDXC Mode	Data Rate	Clock Frequency	Maximum Data Throughput	Actual Throughput
Default Speed/SDR12	Single	0 – 25 MHz	12.5 MB/s	12 MB/s
High Speed/SDR25	Single	0 – 50 MHz	25 MB/s	24 MB/s
SDR50	Single	0 – 100 MHz	50 MB/s	48 MB/s
DDR50	Dual	0 – 50 MHz	50 MB/s	48 MB/s
SDR104	Single	0 – 208 MHz	104 MB/s	96 MB/s

## 37.0 Direct Media Interface (DMI)

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

### 37.1 Signal Description

Name	Type	Description
DMI[7:0]_RXN, DMI[7:0]_RXP	I	DMI receive lanes
DMI[7:0]_TXN, DMI[7:0]_TXP	O	DMI transmit lanes

### 37.2 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/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
<i>Note:</i> 1. Reset reference for primary well pins is RSMRST#.					

### 37.3 Functional Description

The DMI Link supports a maximum of x4 widths.

#### 37.3.1 Lane Reversal

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

#### 37.3.2 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.

## 38.0 Private Configuration Space Target Port ID

The PCH incorporates a wide variety of devices and functions. The registers within these devices are mainly accessed through the primary interface, such as PCI configuration space and IO/MMIO space. Some devices also have registers that are distributed within the PCH Private Configuration Space at individual endpoints which are only accessible through the PCH Sideband Interface. Refer to Volume 2 for registers that reside in the PCH Private Configuration Space (PCR registers).

These PCH Private Configuration Space Registers can be addressed via SBREG\_BAR, Target Port ID, and register offset.

### Acronyms

Acronyms	Description
OTG	On the go

The following table lists the PCH Target Port IDs for PCR register access.

**Table 115. Private Configuration Space Register Target Port IDs**

PCH Device/Function Type	Target Port ID
General Purpose I/O (GPIO) Community 4	6Ah
General Purpose I/O (GPIO) Community 3	6Bh
General Purpose I/O (GPIO) Community 2	6Ch
General Purpose I/O (GPIO) Community 1	6Dh
General Purpose I/O (GPIO) Community 0	6Eh
DCI	B8h
ISH Controller	BEh
Real Time Clock (RTC)	C3h
Processor Interface, 8254 Timer, HPET, APIC	C4h
SMBus	C6h
LPC	C7h
USB 2.0	CAh
UART, I <sup>2</sup> C, GSPI	CBh
FIA Configuration	CFh
SATA	D9h
Integrated Clock Controller (ICC)	DCh
PCIe* Controller #1 (SPA)	80h
PCIe Controller #2 (SPB)	81h
<i>continued...</i>	

<b>PCH Device/Function Type</b>	<b>Target Port ID</b>
PCIe Controller #3 (SPC)	82h
PCIe Controller #4 (SPD)	83h
PCIe Controller #5 (SPE)	84h
PCIe Controller #6 (SPF)	85h
USB Dual Role / OTG	E5h
eSPI / SPI	72h
DMI Configuration	88h
Intel® Trace Hub	B6h
CNVi	73h