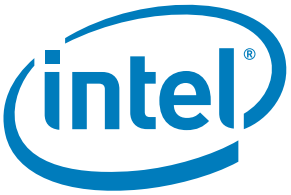


# Intel® 100 Series and Intel® C230 Series Chipset Family Platform Controller Hub (PCH)

## Datasheet – Volume 1 of 2

**December 2018**

Document Number: 332690-005EN

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

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

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

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

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

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

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

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

$I^2C$ is a two-wire communications bus/protocol developed by NXP. SMBus is a subset of the I2C bus/protocol and was developed by Intel. Implementations of the I2C bus/protocol may require licenses from various entities, including NXP Semiconductors N.V.

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

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

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

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

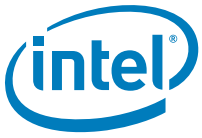
No computer system can provide absolute security under all conditions. Intel® Trusted Execution Technology (Intel® TXT) requires a computer system with Intel® Virtualization Technology, an Intel TXT-enabled processor, chipset, BIOS, Authenticated Code Modules and an Intel TXT-compatible Measured Launched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

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

Intel, and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

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

Copyright © 2016-2018, Intel Corporation



# Contents


**1 Introduction** ..... 23
1.1 About this Manual ..... 23
1.2 References ..... 23
1.3 Overview ..... 23
1.4 PCH SKUs ..... 25

**2 PCH Controller Device IDs** ..... 27
2.1 Device and Revision ID Table ..... 27

**3 Flexible I/O** ..... 32
3.1 Acronyms ..... 32
3.2 References ..... 32
3.3 Overview ..... 32
3.4 Description ..... 32
3.4.1 PCH-H Flexible I/O ..... 33
3.5 HSIO Port Selection ..... 34
3.5.1 PCIe/SATA Port Selection ..... 34

**4 Memory Mapping** ..... 35
4.1 Overview ..... 35
4.2 Functional Description ..... 35
4.2.1 PCI Devices and Functions ..... 35
4.2.2 Fixed I/O Address Ranges ..... 36
4.2.3 Variable I/O Decode Ranges ..... 38
4.3 Memory Map ..... 39
4.3.1 Boot Block Update Scheme ..... 41

**5 System Management** ..... 43
5.1 Acronyms ..... 43
5.2 References ..... 43
5.3 Overview ..... 43
5.4 Features ..... 43
5.4.1 Theory of Operation ..... 44
5.4.1.1 Detecting a System Lockup ..... 44
5.4.1.2 Handling an Intruder ..... 44
5.4.1.3 Detecting Improper Flash Programming ..... 44
5.4.2 TCO Modes ..... 45
5.4.2.1 TCO Compatible Mode ..... 45
5.4.2.2 Advanced TCO Mode ..... 46

**6 High Precision Event Timer (HPET)** ..... 47
6.1 References ..... 47
6.2 Overview ..... 47
6.2.1 Timer Accuracy ..... 47
6.2.2 Timer Off-load ..... 48
6.2.3 Off-loadable Timer ..... 48
6.2.4 Interrupt Mapping ..... 49
6.2.4.1 Mapping Option #1 (Legacy Replacement Option) ..... 49
6.2.4.2 Mapping Option #2 (Standard Option) ..... 49
6.2.4.3 Mapping Option #3 (Processor Message Option) ..... 50
6.2.5 Periodic Versus Non-Periodic Modes ..... 50
6.2.5.1 Non-Periodic Mode ..... 50
6.2.5.2 Periodic Mode ..... 50

6.2.6 Enabling the Timers ....51
6.2.7 Interrupt Levels ....51
6.2.8 Handling Interrupts ....51
6.2.9 Issues Related to 64-Bit Timers with 32-Bit Processors ....51

**7 Thermal Management** ....52
7.1 PCH Thermal Sensor ....52
7.1.1 Modes of Operation ....52
7.1.2 Temperature Trip Point ....52
7.1.3 Thermal Sensor Accuracy (Taccuracy) ....52
7.1.4 Thermal Reporting to an EC ....52
7.1.5 Thermal Trip Signal (PCHHOT#) ....53
7.1.6 Thermal Sensor Programming ....53

**8 Power and Ground Signals** ....54

**9 Pin Straps** ....56

**10 Electrical Characteristics** ....59
10.1 Absolute Maximum Ratings ....59
10.2 Thermal Specification ....59
10.3 PCH Power Supply Range ....60
10.4 General DC Characteristics ....60
10.5 AC Characteristics ....71
10.5.1 Panel Power Sequencing and Backlight Control ....73
10.6 Overshoot/Undershoot Guidelines ....91
10.7 Power Sequencing and Reset Signal Timings ....92

**11 Ballout Definition** ....93

**12 8254 Timers** ....105
12.1 Overview ....105
12.1.1 Timer Programming ....105
12.1.2 Reading from the Interval Timer ....106
12.1.2.1 Simple Read ....106
12.1.2.2 Counter Latch Command ....107
12.1.2.3 Read Back Command ....107

**13 Integrated High Definition Audio** ....108
13.1 Acronyms ....108
13.2 References ....108
13.3 Overview ....108
13.4 Signal Description ....108
13.5 Integrated Pull-Ups and Pull-Downs ....109
13.6 I/O Signal Planes and States ....110
13.7 Features ....110
13.7.1 High Definition Audio Controller Capabilities ....110
13.7.2 Audio DSP Capabilities ....111
13.7.3 High Definition Audio Link Capabilities ....111
13.7.4 Display Audio Link Capabilities ....111
13.7.5 DSP I/O Peripherals Capabilities ....111

**14 Controller Link** ....112
14.1 Overview ....112
14.2 Signal Description ....112
14.3 Integrated Pull-Ups and Pull-Downs ....112
14.4 I/O Signal Planes and States ....112
14.5 Functional Description ....112

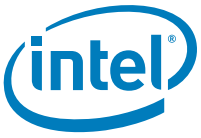
**15 Processor Sideband Signals** ..... 113
15.1 Acronyms ..... 113
15.2 Overview ..... 113
15.3 Signal Description ..... 113
15.4 Integrated Pull-Ups and Pull-Downs ..... 113
15.5 I/O Signal Planes and States ..... 113
15.6 Functional Description ..... 114

**16 Digital Display Signals** ..... 115
16.1 Acronyms ..... 115
16.2 References ..... 115
16.3 Signal Description ..... 115
16.4 Embedded DisplayPort* (eDP*) Backlight Control Signals ..... 116
16.5 Integrated Pull-Ups and Pull-Downs ..... 116
16.6 I/O Signal Planes and States ..... 116

**17 Enhanced Serial Peripheral Interface (eSPI)** ..... 118
17.1 Acronyms ..... 118
17.2 References ..... 118
17.3 Overview ..... 118
17.4 Signal Description ..... 118
17.5 Integrated Pull-Ups and Pull-Downs ..... 119
17.6 I/O Signal Planes and States ..... 119
17.7 Functional Description ..... 120
17.7.1 Features ..... 120
17.7.2 Protocols ..... 120
17.7.3 WAIT States from eSPI Slave ..... 121
17.7.4 In-Band Link Reset ..... 121
17.7.5 Slave Discovery ..... 121
17.7.6 Channels and Supported Transactions ..... 121
17.7.6.1 Peripheral Channel (Channel 0) Overview ..... 122
17.7.6.2 Virtual Wire Channel (Channel 1) Overview ..... 122
17.7.6.3 Out-of-Band Channel (Channel 2) Overview ..... 123
17.7.6.4 Flash Access Channel (Channel 3) Overview ..... 125

**18 General Purpose Input and Output (GPIO)** ..... 127
18.1 Acronyms ..... 127
18.2 References ..... 127
18.3 Overview ..... 127
18.4 Signal Description ..... 128
18.5 Integrated Pull-ups and Pull-downs ..... 139
18.6 Functional Description ..... 139
18.6.1 SMI# / SCI and NMI ..... 139
18.6.2 Blink/PWM Capability ..... 139
18.6.2.1 PWM Programming Sequence ..... 140
18.6.3 Triggering ..... 141
18.6.4 Sx GPIO Implementation Considerations ..... 141
18.6.5 GPIO Ownership ..... 142
18.6.6 GPIO Pad Voltage Tolerance Configuration ..... 142

**19 Intel® Serial I/O Generic SPI (GSPI) Controllers** ..... 143
19.1 Acronyms ..... 143
19.2 References ..... 143
19.3 Overview ..... 143
19.4 Signal Description ..... 143
19.5 Integrated Pull-Ups and Pull-Downs ..... 144
19.6 I/O Signal Planes and States ..... 144




19.7 Functional Description ..... 144
19.7.1 Features ..... 144
19.7.2 Controller Overview ..... 144
19.7.3 DMA Controller ..... 145
19.7.3.1 DMA Transfer and Setup Modes ..... 145
19.7.3.2 Channel Control ..... 145
19.7.4 Reset ..... 146
19.7.5 Power Management ..... 146
19.7.5.1 Device Power Down Support ..... 146
19.7.5.2 Latency Tolerance Reporting (LTR) ..... 146
19.7.6 Interrupts ..... 147
19.7.7 Error Handling ..... 147

**20 Intel® Serial I/O Inter-Integrated Circuit ($I^2C$) Controllers ..... 148**
20.1 Acronyms ..... 148
20.2 References ..... 148
20.3 Overview ..... 148
20.4 Signal Description ..... 148
20.5 Integrated Pull-Ups and Pull-Downs ..... 149
20.6 I/O Signal Planes and States ..... 149
20.7 Functional Description ..... 149
20.7.1 Features ..... 149
20.7.2 Protocols Overview ..... 150
20.7.2.1 Combined Formats ..... 150
20.7.3 DMA Controller ..... 151
20.7.3.1 DMA Transfer and Setup Modes ..... 151
20.7.3.2 Channel Control ..... 151
20.7.4 Reset ..... 151
20.7.5 Power Management ..... 152
20.7.5.1 Device Power Down Support ..... 152
20.7.5.2 Latency Tolerance Reporting (LTR) ..... 152
20.7.6 Interrupts ..... 152
20.7.7 Error Handling ..... 152
20.7.8 Programmable SDA Hold Time ..... 152

**21 Gigabit Ethernet Controller ..... 154**
21.1 Acronyms ..... 154
21.2 References ..... 154
21.3 Overview ..... 154
21.4 Signal Description ..... 154
21.5 Integrated Pull-Ups and Pull-Downs ..... 155
21.6 I/O Signal Planes and States ..... 155
21.7 Functional Description ..... 156
21.7.1 GbE PCI Express* Bus Interface ..... 157
21.7.1.1 Transaction Layer ..... 157
21.7.1.2 Data Alignment ..... 157
21.7.1.3 Configuration Request Retry Status ..... 158
21.7.2 Error Events and Error Reporting ..... 158
21.7.2.1 Completer Abort Error Handling ..... 158
21.7.2.2 Unsupported Request Error Handling ..... 158
21.7.3 Ethernet Interface ..... 158
21.7.3.1 Intel® Ethernet Connection I219 ..... 159
21.7.4 PCI Power Management ..... 159

**22 Interrupt Interface ..... 160**
22.1 Acronyms ..... 160
22.2 References ..... 160

22.3 Overview ..... 160
22.4 Signal Description ..... 160
22.5 Integrated Pull-Ups and Pull-Downs ..... 160
22.6 I/O Signal Planes and States ..... 160
22.7 Functional Description ..... 161
22.7.1 8259 Interrupt Controllers (PIC) ..... 164
22.7.2 Interrupt Handling ..... 165
22.7.2.1 Generating Interrupts ..... 165
22.7.2.2 Acknowledging Interrupts ..... 165
22.7.2.3 Hardware/Software Interrupt Sequence ..... 166
22.7.3 Initialization Command Words (ICWx) ..... 166
22.7.3.1 ICW1 ..... 166
22.7.3.2 ICW2 ..... 167
22.7.3.3 ICW3 ..... 167
22.7.3.4 ICW4 ..... 167
22.7.4 Operation Command Words (OCW) ..... 167
22.7.5 Modes of Operation ..... 167
22.7.5.1 Fully-Nested Mode ..... 167
22.7.5.2 Special Fully-Nested Mode ..... 168
22.7.5.3 Automatic Rotation Mode (Equal Priority Devices) ..... 168
22.7.5.4 Specific Rotation Mode (Specific Priority) ..... 168
22.7.5.5 Poll Mode ..... 168
22.7.5.6 Edge and Level Triggered Mode ..... 169
22.7.5.7 End Of Interrupt (EOI) Operations ..... 169
22.7.5.8 Normal End of Interrupt ..... 169
22.7.5.9 Automatic End of Interrupt Mode ..... 169
22.7.6 Masking Interrupts ..... 170
22.7.6.1 Masking on an Individual Interrupt Request ..... 170
22.7.6.2 Special Mask Mode ..... 170
22.7.7 Steering PCI Interrupts ..... 170
22.8 Advanced Programmable Interrupt Controller (APIC) (D31:F0) ..... 170
22.8.1 Interrupt Handling ..... 170
22.8.2 Interrupt Mapping ..... 171
22.8.3 PCI/PCI Express* Message-Based Interrupts ..... 172
22.8.4 IOxAPIC Address Remapping ..... 172
22.8.5 External Interrupt Controller Support ..... 172
22.9 Serial Interrupt ..... 172
22.9.1 Start Frame ..... 173
22.9.2 Stop Frame ..... 173
22.9.3 Specific Interrupts Not Supported Using SERIRQ ..... 174

**23 Integrated Sensor Hub (ISH)** ..... 175
23.1 Acronyms ..... 175
23.2 References ..... 175
23.3 Overview ..... 175
23.4 Signal Description ..... 176
23.5 Integrated Pull-Ups and Pull-Downs ..... 176
23.6 I/O Signal Planes and States ..... 176
23.7 Functional Description ..... 177
23.7.1 ISH Micro-Controller ..... 177
23.7.2 SRAM ..... 177
23.7.3 PCI Host Interface ..... 177
23.7.3.1 MMIO Space ..... 177
23.7.3.2 DMA Controller ..... 177
23.7.3.3 PCI Interrupts ..... 178
23.7.3.4 PCI Power Management ..... 178
23.7.4 Power Domains and Management ..... 178

23.7.4.1 ISH Power Management ..... 178
23.7.4.2 External Sensor Power Management ..... 178
23.7.5 ISH IPC ..... 178
23.7.6 ISH Interrupt Handling via IOAPIC (Interrupt Controller) ..... 178
23.7.7 ISH I2C Controllers ..... 179
23.7.8 ISH UART Controller ..... 179
23.7.9 ISH GPIOs ..... 179
23.8 Embedded Location (Comms Hub) ..... 179

**24 Low Pin Count (LPC)** ..... 181
24.1 Acronyms ..... 181
24.2 References ..... 181
24.3 Overview ..... 181
24.4 Signal Description ..... 182
24.5 Integrated Pull-Ups and Pull-Downs ..... 182
24.6 I/O Signal Planes and States ..... 182
24.7 Functional Description ..... 182
24.7.1 LPC Cycle Types ..... 182
24.7.2 Start Field Definition ..... 183
24.7.3 Cycle Type/Direction (CYCTYPE + DIR) ..... 183
24.7.4 Size ..... 183
24.7.4.1 SYNC ..... 184
24.7.5 SYNC Timeout ..... 184
24.7.6 SYNC Error Indication ..... 184
24.7.7 LFRAME# Usage ..... 184
24.7.8 I/O Cycles ..... 185
24.7.9 LPC Power Management ..... 185
24.7.9.1 LPCPD# Protocol ..... 185
24.7.10 Configuration and PCH Implications ..... 185
24.7.10.1 LPC I/F Decoders ..... 185

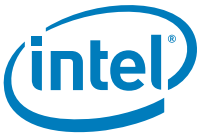
**25 PCH and System Clocks** ..... 186
25.1 Overview ..... 186
25.2 Signal Descriptions ..... 186
25.3 I/O Signal Planes and States ..... 187
25.4 General Features ..... 188

**26 PCI Express* (PCIe*)** ..... 189
26.1 References ..... 189
26.2 Overview ..... 189
26.3 Signal Description ..... 190
26.4 I/O Signal Planes and States ..... 190
26.5 PCI Express* Port Support Feature Details ..... 190
26.5.1 Intel® Rapid Storage Technology (Intel® RST) for PCIe* Storage ..... 191
26.5.1.1 Supported Features Summary ..... 192
26.5.2 Interrupt Generation ..... 192
26.5.3 Power Management ..... 192
26.5.3.1 S3/S4/S5 Support ..... 192
26.5.3.2 Resuming from Suspended State ..... 193
26.5.3.3 Device Initiated PM_PME Message ..... 193
26.5.3.4 SMI/SCI Generation ..... 193
26.5.3.5 Latency Tolerance Reporting (LTR) ..... 194
26.5.4 Dynamic Link Throttling ..... 194
26.5.5 Port 8xh Decode ..... 195
26.5.6 Separate Reference Clock with Independent SSC (SRIS) ..... 195
26.5.7 SERR# Generation ..... 195

26.5.8 Hot-Plug ..... 196
26.5.8.1 Presence Detection ..... 196
26.5.8.2 SMI/SCI Generation ..... 196
26.5.9 PCI Express* Lane Polarity Inversion ..... 196
26.5.10 PCI Express* Controller Lane Reversal ..... 197

**27 Power Management** ..... 198
27.1 Acronyms ..... 198
27.2 References ..... 198
27.3 Overview ..... 198
27.4 Signal Description ..... 198
27.5 Integrated Pull-Ups and Pull-Downs ..... 201
27.6 I/O Signal Planes and States ..... 201
27.7 Functional Description ..... 202
27.7.1 Features ..... 202
27.7.2 PCH and System Power States ..... 203
27.7.3 System Power Planes ..... 205
27.7.4 SMI#/SCI Generation ..... 205
27.7.4.1 PCI Express* SCI ..... 207
27.7.4.2 PCI Express* Hot-Plug ..... 207
27.7.5 C-States ..... 207
27.7.6 Dynamic 24-MHz Clock Control ..... 208
27.7.6.1 Conditions for Checking the 24-MHz Clock ..... 208
27.7.6.2 Conditions for Maintaining the 24-MHz Clock ..... 208
27.7.6.3 Conditions for Stopping the 24-MHz Clock ..... 208
27.7.6.4 Conditions for Re-starting the 24-MHz Clock ..... 208
27.7.7 Sleep States ..... 209
27.7.7.1 Sleep State Overview ..... 209
27.7.7.2 Initiating Sleep State ..... 209
27.7.7.3 Exiting Sleep States ..... 209
27.7.7.4 PCI Express* WAKE# Signal and PME Event Message ..... 211
27.7.7.5 Sx-G3-Sx, Handling Power Failures ..... 211
27.7.7.6 Deep Sx ..... 212
27.7.8 Event Input Signals and Their Usage ..... 213
27.7.8.1 PWRBTN# (Power Button) ..... 213
27.7.8.2 PME# (PCI Power Management Event) ..... 215
27.7.8.3 SYS_RESET# Signal ..... 215
27.7.8.4 THERMTRIP# Signal ..... 215
27.7.8.5 Sx_Exit_Holdoff# ..... 216
27.7.9 ALT Access Mode ..... 216
27.7.9.1 Write Only Registers with Read Paths in ALT Access Mode ..... 217
27.7.9.2 PIC Reserved Bits ..... 218
27.7.9.3 Read Only Registers with Write Paths in ALT Access Mode ..... 218
27.7.10 System Power Supplies, Planes, and Signals ..... 218
27.7.10.1 Power Plane Control ..... 218
27.7.10.2 SLP_S4# and Suspend-to-RAM Sequencing ..... 219
27.7.10.3 PCH_PWROK Signal ..... 219
27.7.10.4 BATLOW# (Battery Low) ..... 219
27.7.10.5 SLP_LAN# Pin Behavior ..... 219
27.7.10.6 SLP_WLAN# Pin Behavior ..... 222
27.7.10.7 SUSPWRDNACK/SUSWARN#/GPP_A13 Steady State Pin Behavior ..... 222
27.7.10.8 RTCRST# and SRTCRST# ..... 223
27.7.11 Legacy Power Management Theory of Operation ..... 223
27.7.11.1 Mobile APM Power Management ..... 223
27.7.12 Reset Behavior ..... 223

**28 Real Time Clock (RTC)** ..... 226
28.1 Acronyms ..... 226




28.2 References ..... 226
28.3 Overview ..... 226
28.4 Signal Description ..... 226
28.5 Integrated Pull-Ups and Pull-Downs ..... 227
28.6 I/O Signal Planes and States ..... 227
28.7 Functional Description ..... 227
28.7.1 Update Cycles ..... 228
28.7.2 Interrupts ..... 228
28.7.3 Lockable RAM Ranges ..... 228
28.7.4 Century Rollover ..... 228
28.7.5 Clearing Battery-Backed RTC RAM ..... 229
28.7.5.1 Using RTCRST# to Clear CMOS ..... 229
28.7.5.2 Using a GPI to Clear CMOS ..... 229
28.7.6 External RTC Circuitry ..... 229

**29 Serial ATA (SATA) ..... 230**
29.1 Acronyms ..... 230
29.2 References ..... 230
29.3 Overview ..... 230
29.4 Signal Description ..... 231
29.5 Integrated Pull-Ups and Pull-Downs ..... 237
29.6 I/O Signal Planes and States ..... 238
29.7 Functional Description ..... 238
29.7.1 SATA 6 Gb/s Support ..... 239
29.7.2 SATA Feature Support ..... 239
29.7.3 Hot-Plug Operation ..... 239
29.7.4 Intel® Rapid Storage Technology (Intel® RST) ..... 239
29.7.4.1 Intel® Rapid Storage Technology (Intel® RST) Configuration ..... 240
29.7.4.2 Intel® Rapid Storage Technology (Intel® RST) RAID Option ROM ..... 240
29.7.5 Intel® Rapid Storage Technology enterprise (Intel® RSTe) - for Server/ Workstation Only ..... 241
29.7.5.1 Intel® Rapid Storage Technology enterprise (Intel® RSTe) Configuration - for Server/Workstation Only ..... 241
29.7.5.2 Intel® Rapid Storage Technology enterprise (Intel® RSTe) Legacy RAID Option ROM - for Server/Workstation Only ..... 242
29.7.5.3 Intel® Rapid Storage Technology enterprise (Intel® RSTe) EFI Driver - for Server/Workstation Only ..... 242
29.7.6 Intel® Smart Response Technology ..... 242
29.7.7 Power Management Operation ..... 243
29.7.7.1 Power State Mappings ..... 243
29.7.7.2 Power State Transitions ..... 243
29.7.7.3 Low Power Platform Consideration ..... 245
29.7.8 SATA Device Presence ..... 245
29.7.9 SATA LED ..... 245
29.7.10 Advanced Host Controller Interface (AHCI) Operation ..... 246
29.7.11 External SATA ..... 246
29.7.12 Enclosure Management (SGPIO Signals) ..... 246
29.7.12.1 Mechanism ..... 247
29.7.12.2 Message Format ..... 248
29.7.12.3 LED Message Type ..... 248
29.7.12.4 SGPIO Waveform ..... 249

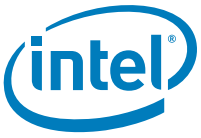
**30 System Management Interface and SMLink ..... 250**
30.1 Acronyms ..... 250
30.2 References ..... 250
30.3 Overview ..... 250
30.4 Signal Description ..... 250

30.5 Integrated Pull-Ups and Pull-Downs ..... 251
30.6 I/O Signal Planes and States ..... 251
30.7 Functional Description ..... 251

**31 Host System Management Bus (SMBus) Controller** ..... 252
31.1 Acronyms ..... 252
31.2 References ..... 252
31.3 Overview ..... 252
31.4 Signal Description ..... 252
31.5 Integrated Pull-Ups and Pull-Downs ..... 252
31.6 I/O Signal Planes and States ..... 253
31.7 Functional Description ..... 253
31.7.1 Host Controller ..... 253
31.7.1.1 Host Controller Operation Overview ..... 253
31.7.1.2 Command Protocols ..... 254
31.7.1.3 Bus Arbitration ..... 258
31.7.1.4 Clock Stretching ..... 258
31.7.1.5 Bus Timeout (PCH as SMBus Master) ..... 258
31.7.1.6 Interrupts/SMI# ..... 258
31.7.1.7 SMBus CRC Generation and Checking ..... 259
31.7.2 SMBus Slave Interface ..... 260
31.7.2.1 Format of Slave Write Cycle ..... 261
31.7.2.2 Format of Read Command ..... 262
31.7.2.3 Slave Read of RTC Time Bytes ..... 264
31.7.2.4 Format of Host Notify Command ..... 264
31.7.2.5 Format of Read Command ..... 265

**32 Serial Peripheral Interface (SPI)** ..... 268
32.1 Acronyms ..... 268
32.2 References ..... 268
32.3 Overview ..... 268
32.4 Signal Description ..... 268
32.5 Integrated Pull-Ups and Pull-Downs ..... 269
32.6 I/O Signal Planes and States ..... 269
32.7 Functional Description ..... 269
32.7.1 SPI for Flash ..... 269
32.7.1.1 Overview ..... 269
32.7.1.2 SPI Supported Features ..... 270
32.7.1.3 Flash Descriptor ..... 271
32.7.1.4 Flash Access ..... 273
32.7.2 SPI Support for TPM ..... 274

**33 Testability** ..... 275
33.1 JTAG ..... 275
33.1.1 Acronyms ..... 275
33.1.2 References ..... 275
33.1.3 Overview ..... 275
33.1.4 Signal Description ..... 275
33.1.5 I/O Signal Planes and States ..... 276
33.2 Intel® Trace Hub (Intel® TH) ..... 276
33.2.1 Overview ..... 276
33.2.2 Platform Setup ..... 277
33.3 Direct Connect Interface (DCI) ..... 277
33.3.1 Boundary Scan Side Band (BSSB) Hosting DCI ..... 278
33.3.2 USB3 Hosting DCI ..... 278
33.3.3 Platform Setup ..... 279




**34 Intel® Serial I/O Universal Asynchronous Receiver/Transmitter (UART) Controllers .. 280**

34.1 Acronyms ..... 280
34.2 References ..... 280
34.3 Overview ..... 280
34.4 Signal Description ..... 280
34.5 Integrated Pull-Ups and Pull-Downs ..... 281
34.6 I/O Signal Planes and States ..... 281
34.7 Functional Description ..... 281
34.7.1 Features ..... 281
34.7.2 UART Serial (RS-232) Protocols Overview ..... 282
34.7.3 16550 8-bit Addressing - Debug Driver Compatibility ..... 283
34.7.4 DMA Controller ..... 283
34.7.4.1 DMA Transfer and Setup Modes ..... 283
34.7.4.2 Channel Control ..... 283
34.7.5 Reset ..... 284
34.7.6 Power Management ..... 284
34.7.6.1 Device Power Down Support ..... 284
34.7.6.2 Latency Tolerance Reporting (LTR) ..... 284
34.7.7 Interrupts ..... 285
34.7.8 Error Handling ..... 285

**35 Universal Serial Bus (USB) ..... 286**

35.1 Acronyms ..... 286
35.2 References ..... 286
35.3 Overview ..... 286
35.4 Signal Description ..... 286
35.5 Integrated Pull-Ups and Pull-Downs ..... 289
35.6 I/O Signal Planes and States ..... 289
35.7 Functional Description ..... 290
35.7.1 eXtensible Host Controller Interface (xHCI) Controller (D20:F0) ..... 290
35.7.1.1 USB Dual Role Support ..... 290

**36 GPIO Serial Expander ..... 291**

36.1 Acronyms, Definitions ..... 291
36.2 References ..... 291
36.3 Overview ..... 291
36.4 Signal Description ..... 291
36.5 Integrated Pull-ups and Pull-downs ..... 291
36.6 Functional Description ..... 291

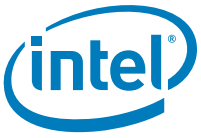
**37 Fan Control ..... 293**

37.1 Acronyms ..... 293
37.2 References ..... 293
37.3 Overview ..... 293
37.4 Signal Description ..... 293
37.5 Integrated Pull-ups and Pull-downs ..... 294
37.6 I/O Signal Planes and States ..... 294
37.7 Functional Description ..... 294

**38 Direct Media Interface ..... 295**

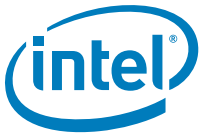
38.1 Acronyms ..... 295
38.2 References ..... 295
38.3 Overview ..... 295
38.4 Signal Description ..... 295
38.5 Integrated Pull-ups and Pull-downs ..... 295
38.6 I/O Signal Planes and States ..... 296

38.7 Functional Description ..... 296
**39 Primary to Sideband Bridge (P2SB)** ..... 297
39.1 Overview ..... 297



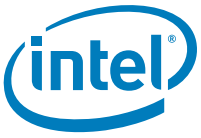
# Figures


3-1 HSIO Multiplexing on PCH-H ..... 29
5-1 TCO Compatible Mode SMBus Configuration ..... 41
5-2 Advanced TCO Mode ..... 42
10-2 PCI Express* Receiver Eye ..... 69
10-1 PCI Express* Transmitter Eye ..... 69
10-3 Panel Power Sequencing ..... 70
10-4 Clock Timing ..... 73
10-5 Measurement Points for Differential Waveforms ..... 74
10-6 SMBus/SMLink Transaction ..... 75
10-7 PCH Test Load ..... 75
10-8 USB Rise and Fall Times ..... 77
10-9 USB Jitter ..... 77
10-10 USB EOP Width ..... 77
10-11 SMBus/SMLink Timeout ..... 79
10-12 Intel® High Definition Audio (Intel® HD Audio) Input and Output Timings ..... 80
10-13 Valid Delay from Rising Clock Edge ..... 80
10-14 Setup and Hold Times ..... 81
10-15 Float Delay ..... 81
10-16 Output Enable Delay ..... 81
10-17 Valid Delay from Rising Clock Edge ..... 82
10-18 Setup and Hold Times ..... 82
10-19 Pulse Width ..... 82
10-20 SPI Timings ..... 84
10-21 GSPI Timings ..... 85
10-22 Controller Link Receive Timings ..... 86
10-23 Controller Link Receive Slew Rate ..... 86
10-24 Maximum Acceptable Overshoot/Undershoot Waveform ..... 88
11-1 BGA PCH-H Ballout (Top View – Upper Left) ..... 90
11-2 BGA PCH-H Ballout (Top View – Upper Right) ..... 91
11-3 BGA PCH-H Ballout (Top View – Lower Left) ..... 92
11-4 BGA PCH-H Ballout (Top View – Lower Right) ..... 93
17-1 Basic eSPI Protocol ..... 116
17-2 eSPI Slave Request to PCH for PCH Temperature ..... 119
17-3 PCH Response to eSPI Slave with PCH Temperature ..... 120
17-4 eSPI Slave Request to PCH for PCH RTC Time ..... 120
17-5 PCH Response to eSPI Slave with RTC Time ..... 121
20-1 Data Transfer on the $I^2C$ Bus ..... 146
24-1 LPC Interface Diagram ..... 177
25-1 PCH Internal Clock Diagram ..... 182
26-1 Generation of SERR# to Platform ..... 192
27-1 Conceptual Diagram of SLP_LAN# ..... 217
29-1 Flow for Port Enable/Device Present Bits ..... 241
29-2 Serial Data transmitted over the SGPIO Interface ..... 245
32-1 Flash Descriptor Regions ..... 268
33-1 Platform Setup with Intel® Trace Hub ..... 273
33-2 Platform Setup with DCI Connection ..... 274
34-1 UART Serial Protocol ..... 278
34-2 UART Receiver Serial Data Sample Points ..... 278
36-1 Example of GSX Topology ..... 288



# Tables


1-1 PCH-H I/O Capabilities ..... 21
1-2 PCH-H SKUs ..... 22
1-3 PCH-H HSIO Detail (Lane 1-14) ..... 22
1-4 PCH-H HSIO Detail (Lane 15-26) ..... 22
1-5 Mobile Client/Server/Mobile SKUs ..... 23
1-6 Mobile Client/Server/Workstation SKUs ..... 23
1-7 Mobile Client /Server/Workstation PCH HSIO Detail (Lane 1-13) ..... 24
1-8 Mobile Client/Server/Workstation PCH HSIO Detail (Lane 14-26) ..... 24
2-1 PCH-H Device and Revision ID Table ..... 25
4-1 PCI Devices and Functions ..... 31
4-2 Fixed I/O Ranges Decoded by PCH ..... 33
4-3 Variable I/O Decode Ranges ..... 35
4-4 PCH Memory Decode Ranges (Processor Perspective) ..... 35
4-5 SPI Mode Address Swapping ..... 38
5-1 Event Transitions that Cause Messages ..... 41
6-1 Legacy Replacement Routing ..... 45
9-1 Functional Strap Definitions ..... 52
10-1 PCH Absolute Maximum Ratings ..... 55
10-2 Thermal Design Power ..... 55
10-3 PCH Power Supply Range ..... 56
10-4 PCH-H Measured $I_{CC}$ (Desktop and Server SkUs) ..... 56
10-5 PCH-H Measured $I_{CC}$ (H Mobile SkUs) ..... 57
10-6 PCH-H VCCMPHY_1p0 Icc Adder Per HSIO Lane ..... 58
10-7 Single-Ended Signal DC Characteristics as Inputs or Outputs ..... 59
10-8 Single-Ended Signal DC Characteristics as Inputs or Outputs ..... 63
10-9 Differential Signals Characteristics ..... 63
10-10 Other DC Characteristics ..... 66
10-11 PCI Express* Interface Timings ..... 67
10-12 DDC Characteristics ..... 70
10-13 DisplayPort* Hot-Plug Detect Interface ..... 71
10-14 Clock Timings ..... 71
10-15 USB 2.0 Timing ..... 75
10-16 USB 3.0 Interface Transmit and Receiver Timings ..... 76
10-17 SATA Interface Timings ..... 77
10-18 SMBus and SMLink Timing ..... 78
10-19 Intel® High Definition Audio (Intel® HD Audio) Timing ..... 79
10-20 LPC Timing ..... 80
10-21 Miscellaneous Timings ..... 81
10-22 SPI Timings (17MHz) ..... 82
10-23 SPI Timings (30 MHz) ..... 83
10-24 SPI Timings (48 MHz) ..... 83
10-25 GSPI Timings (20 MHz) ..... 84
10-26 Controller Link Receive Timings ..... 85
10-27 UART Timings ..... 86
10-28 $I^2S$ Timings ..... 86
10-29 3.3V Overshoot/Undershoot Specifications ..... 87
10-30 1.8V Overshoot/Undershoot Specifications ..... 88
11-1 BGA PCH-H Ballout ..... 94
12-1 Counter Operating Modes ..... 102
13-1 Integrated Pull-Ups and Pull-Downs ..... 105
13-2 I/O Signal Planes and States ..... 106
16-1 Digital Display Signals ..... 111
17-1 eSPI Channels and Supported Transactions ..... 117



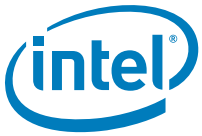

17-2 eSPI Virtual Wires (VW) ..... 118
18-1 GPIO Group Summary ..... 123
18-2 General Purpose I/O Signals ..... 124
18-3 PWM Output Frequencies Assuming 32.768 KHz ..... 136
21-1 GbE LAN Signals ..... 150
21-2 Integrated Pull-Ups and Pull-Downs ..... 151
21-3 Power Plane and States for Output Signals ..... 151
21-4 Power Plane and States for Input Signals ..... 151
21-5 LAN Mode Support ..... 155
22-1 Interrupt Options - 8259 Mode ..... 157
22-2 Interrupt Options - APIC Mode ..... 158
22-3 Interrupt Logic Signals ..... 159
22-4 Interrupt Controllers PIC ..... 160
22-5 Interrupt Status Registers ..... 161
22-6 Content of Interrupt Vector Byte ..... 161
22-7 APIC Interrupt Mapping1 ..... 167
22-8 Stop Frame Explanation ..... 169
22-9 Data Frame Format ..... 170
23-1 IPC Initiator -> Target flows ..... 174
24-1 LPC Cycle Types Supported ..... 178
24-2 Start Field Bit Definitions ..... 179
24-3 Cycle Type Bit Definitions ..... 179
24-4 Transfer Size Bit Definition ..... 180
24-5 SYNC Bit Definition ..... 180
25-1 I/O Signal Planes and States ..... 183
26-1 PCI Express* Port Feature Details ..... 186
26-2 PCI Express* Link Configurations Supported ..... 187
26-3 MSI Versus PCI IRQ Actions ..... 188
27-1 General Power States for Systems Using the PCH ..... 199
27-2 State Transition Rules for the PCH ..... 200
27-3 System Power Plane ..... 201
27-4 Causes of SMI and SCI ..... 202
27-5 Sleep Types ..... 205
27-6 Causes of Wake Events ..... 206
27-7 Transitions Due to Power Failure ..... 207
27-8 Supported Deep Sx Policy Configurations ..... 208
27-9 Deep Sx Wake Events ..... 209
27-10 Transitions Due to Power Button ..... 209
27-11 Write Only Registers with Read Paths in ALT Access Mode ..... 213
27-12 PIC Reserved Bits Return Values ..... 214
27-13 Register Write Accesses in ALT Access Mode ..... 214
27-14 SUSPWRDNACK/SUSWARN#/GPP_A13 Pin Behavior ..... 218
27-15 SUSPWRDNACK During Reset ..... 218
27-16 Causes of Host and Global Resets ..... 220
28-1 RTC Crystal Requirements ..... 225
28-2 External Crystal Oscillator Requirements ..... 225
31-1 $I^2C$* Block Read ..... 252
31-2 Enable for SMBALERT# ..... 255
31-3 Enables for SMBus Slave Write and SMBus Host Events ..... 255
31-4 Enables for the Host Notify Command ..... 255
31-5 Slave Write Registers ..... 257
31-6 Command Types ..... 257
31-7 Slave Read Cycle Format ..... 258
31-8 Data Values for Slave Read Registers ..... 258
31-9 Host Notify Format ..... 261

31-10 Slave Read Cycle Format ..... 261
31-11 Data Values for Slave Read Registers ..... 262
31-12 Enables for SMBus Slave Write and SMBus Host Events ..... 263
32-1 SPI Flash Regions ..... 266
32-2 Region Size Versus Erase Granularity of Flash Components ..... 267
32-3 Region Access Control Table ..... 269
39-1 Private Configuration Space Register Target Port IDs ..... 293

# Revision History

| Revision Number | Description | Date |
|---|---|---|
| 001 | • Initial Release | August 2015 |
| 002 | • Added HM170, QM170 and CM236 Mobile SKUs and H170, B150, H110, Q170 and Q150 Desktop SKUs<br>**Chapter 1, "Introduction"**<br>• Updated Table 1-2, PCH-H SKUs"<br>**Chapter 11, "Ballout Definition"**<br>• Added Chapter 11.<br>**Chapter 16, "Digital Display Signals"**<br>• Updated Section 16.5 and 16.6. Added DDPD_CTRLDATA and DDPD_CTRCLK to the tables. | October 2015 |
| 003 | **Chapter 32, "Serial Peripheral Interface (SPI)"**<br>• Updated Section 32.5<br>• Updated Section 32.6<br>**Chapter 34, "Intel® Serial I/O Universal Asynchronous Receiver/Transmitter (UART) Controllers"**<br>• Updated Section 34.3 | February 2016 |
| 004 | **Chapter 4, "Memory Mapping"**<br>• Removed Serial Port-3 in Table 4-3<br>**Chapter 18, "General Purpose Input and Output (GPIO)"**<br>• Updated Table 18-2 Note column. Changed PLTRST# to PCH_PWROK for GPP_B14, GPP_B18, GPP_B22, GPP_I6, GPP_I8 and GPP_I10<br>**Chapter 20, "Intel® Serial I/O Inter-Integrated Circuit (I2C) Controllers"**<br>• Updated Section 20.7.1<br>**Chapter 23, "Integrated Sensor Hub (ISH)"**<br>• Typo correction on ISH UART 1 signals in Section 23.4<br>**Chapter 32, "Serial Peripheral Interface (SPI)"**<br>• Updated PU/PD info for SPI CKL and CS# signals in Section 32.5 and 32.6<br>**Chapter 34, "Intel® Serial I/O Universal Asynchronous Receiver/Transmitter (UART) Controllers"**<br>• Updated Section 34.3 | May 2016 |
| 005 | **Chapter 2, "PCH Controller Device IDs"**<br>• Updated Table 2-1<br>**Chapter 10, "Electrical Characteristics"**<br>• Updated Table 10-7 | December 2018 |

§ §



# 1 Introduction

## 1.1 About this Manual

This document is intended for Original Equipment Manufacturers (OEMs), Original Design Manufacturers (ODM) and BIOS vendors creating products based on the Intel® 100 Series and Intel® C230 Series Chipset Family Platform Controller Hub (PCH).

***Note:*** Throughout this document, the Platform Controller Hub (PCH) is used as a general term and refers to all Intel® 100 Series and Intel® C230 Series PCH SKUs, unless specifically noted otherwise.

***Note:*** Throughout this document, PCH-H refers to desktop, server/workstation, and mobile segment PCH SKUs, unless specifically noted otherwise.

***Note:*** Throughout this document, the terms "Desktop" and "Desktop Only" refers to information that is applicable only to Desktop PCH, unless specifically noted otherwise.

***Note:*** 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.

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

## 1.2 References

| Specification | Document #/Location |
|---|---|
| Intel® 100 Series and Intel® C230 Series Chipset Family Platform Controller Hub (PCH) Datasheet, Volume 2 of 2 | 332691-00EN |

## 1.3 Overview

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

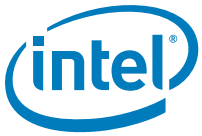
- ACPI Power Management Logic Support, Revision 4.0a
- PCI Express* Base Specification Revision 3.0
- Integrated Serial ATA Host controller, supports data transfer rates of up to 6Gb/s on all ports
- xHCI USB controller with SuperSpeed USB 3.0 ports

- USB Dual Role/OTG Capability
- 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 or USB 3.0
- General Purpose Input Output (GPIO)
- Low Pin Count (LPC) interface
- Interrupt controller
- Timer functions
- System Management Bus (SMBus) Specification, Version 2.0
- Integrated Clock Controller (ICC)/Real Time Clock Controller (RTCC)
- Intel® High Definition Audio and Intel® Smart Sound Technology (Intel® SST)
- Intel® Serial I/O UART Host controllers
- Intel® Serial I/O $I^2C$ Host controllers
- Integrated 10/100/1000 Gigabit Ethernet MAC
- Integrated Sensor Hub (ISH)
- Supports Intel® Rapid Storage Technology (Intel® RST)
- Supports Intel® Active Management Technology (Intel® AMT)
- Supports Intel® Virtualization Technology for Directed I/O (Intel® VT-d)
- Supports Intel® Trusted Execution Technology (Intel® TXT)
- JTAG Boundary Scan support
- Intel® Trace Hub (Intel® TH) and Direct Connect Interface (DCI) for debug

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

**Table 1-1. PCH-H I/O Capabilities**

| Interface | PCH-H |
|---|---|
| CPU Interface | DMI Gen3 x4 |
| PCIe | Up to 20 Gen3 lanes (up to 16 devices max) |
| USB | Up to 10 SS, 14 HS, 1 OTG |
| SATA | Up to 6 SATA Revision 3.0 |
| LAN Ports | 1 GBE |
| Audio | Intel® HD Audio, $I^2S$ (Bluetooth), Direct attach Digital Mic (DMIC) |
| LPC | 24 MHz, No DMA |
| eSPI | 1 CS#, Quad Mode |
| $I^2C$ | 2 |
| UART | 3 |
| Generic SPI (GSPI) | 2 |
| Integrated Sensor Hub (ISH) | 2 $I^2C$, 2 UART |



# 1.4 PCH SKUs

## Table 1-2. PCH-H SKUs

| Features | H110 | H170 | HM170 | QM170 | Z170 | B150 | Q150 | Q170 |
|---|---|---|---|---|---|---|---|---|
| Intel® Rapid Storage Technology | AHCI Mode | Full Features[8] | Full Features[8] | Full Features[8] | Full Features[8] | AHCI Mode | AHCI Mode | Full Features[8] |
| Total USB 3.0 Ports | 4 | 8 | 8 | 8 | 10 | 6 | 8 | 10 |
| Total USB 2.0 Ports | 10[1] | 14[3] | 14[3] | 14[3] | 14[3] | 12[2] | 14[3] | 14[3] |
| Total SATA 3.0 Ports (Max 6 Gb/s) | 4 | 6 | 4 | 4 | 6 | 6 | 6 | 6 |
| Total PCI Express* Lanes (Gen) | 6 (2.0) | 16 (3.0) | 16 (3.0) | 16 (3.0) | 20 (3.0) | 8 (3.0) | 10 (3.0) | 20 (3.0) |
| Total Intel® RST capable PCIe and SATA Express[4] Storage Devices | 0 | 2[6] | 2[6] | 2[6] | 3[7] | 0 | 0 | 3[7] |
| Processor dgfx bifurcation support | No | No | Yes[5] | Yes[5] | Yes[5] | No | No | Yes[5] |

***Notes:***
1. USB 2.0 port numbers: 1-10
2. USB 2.0 port numbers: 1-12
3. USB 2.0 port numbers: 1-14
4. SATA Express Capable Ports (x2)
5. PCIe configuration 1x16, or 2x8 or 2x4 or 1x8 are supported
6. Intel® RST PCIe supports RAID configuration 0/1.
7. Intel® RST PCIe supports RAID configuration 0/1/5
8. Full featured includes SATA RAID 0/1/5/10 support

## Table 1-3. PCH-H HSIO Detail (Lane 1-14)

| SKU | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| H110 | USB 3.0/ OTG | USB 3.0 | USB 3.0 | USB 3.0 | N/A | N/A | N/A | N/A | N/A | LAN Only | PCIe/ LAN | PCIe | PCIe | PCIe |
| H170 | USB 3.0/ OTG | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | PCIe | PCIe/ LAN | PCIe/ LAN | PCIe | PCIe | PCIe |
| HM170 | USB 3.0/ OTG | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0/ PCIe | USB 3.0/ PCIe | PCIe | PCIe / LAN | PCIe/ LAN | PCIe | PCIe | PCIe |
| QM170 | USB 3.0/ OTG | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0/ PCIe | USB 3.0/ PCIe | PCIe | PCIe / LAN | PCIe/ LAN | PCIe | PCIe | PCIe |
| Z170 | USB 3.0/ OTG | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0/ PCIe | USB 3.0/ PCIe | USB 3.0/ PCIe | USB 3.0/ PCIe/LAN | PCIe/ LAN | PCIe | PCIe | PCIe |
| B150 | USB 3.0/ OTG | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | N/A | N/A | N/A | LAN Only | PCIe/ LAN | PCIe | PCIe | PCIe |
| Q150 | USB 3.0/ OTG | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | N/A | LAN Only | PCIe/ LAN | PCIe | PCIe | PCIe |
| Q170 | USB 3.0/ OTG | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0/ PCIe | USB 3.0/ PCIe | USB 3.0/ PCIe | USB 3.0/ PCIe/LAN | PCIe / LAN | PCIe | PCIe | PCIe |

## Table 1-4. PCH-H HSIO Detail (Lane 15-26) (Sheet 1 of 2)

| SKU | 15[1] | 16[1] | 17 | 18 | 19[1] | 20[1] | 21 | 22 | 23 | 24 | 25 | 26 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| H110 | PCIe/ LAN | PCIe | N/A | LAN Only | SATA0/ LAN | SATA1 | SATA | SATA | N/A | N/A | N/A | N/A |
| H170 | PCIe/LAN /SATA0 | PCIe/ SATA1 | PCIe | PCIe/ LAN | PCIe/LAN / SATA0 | PCIe/ SATA1 | PCIe/ SATA | PCIe/ SATA | SATA | SATA | PCIe | PCIe |
| HM170 | PCIe/LAN /SATA0 | PCIe/LAN/ SATA1 | PCIe | PCIe/ LAN | PCIe/LAN/ SATA0 | PCIe/ SATA1 | PCIe/ SATA | PCIe/ SATA | N/A | N/A | N/A | N/A |
| QM170 | PCIe/LAN /SATA0 | PCIe/LAN/ SATA1 | PCIe | PCIe/ LAN | PCIe/ LAN/ SATA0 | PCIe/ SATA1 | PCIe/ SATA | PCIe/ SATA | N/A | N/A | N/A | N/A |
| Z170 | PCIe/LAN /SATA0 | PCIe/ SATA1 | PCIe | PCIe/ LAN | PCIe/LAN / SATA0 | PCIe/ SATA1 | PCIe/ SATA | PCIe/ SATA | PCIe/ SATA | PCIe/ SATA | PCIe | PCIe |
| B150 | PCIe/LAN /SATA0 | PCIe/ SATA1 | PCIe | PCIe/ LAN | SATA0 | SATA1 | SATA | SATA | SATA | SATA | N/A | N/A |

## Table 1-4. PCH-H HSIO Detail (Lane 15-26) (Sheet 2 of 2)

| SKU | 15[1] | 16[1] | 17 | 18 | 19[1] | 20[1] | 21 | 22 | 23 | 24 | 25 | 26 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Q150 | PCIe/LAN /SATA0 | PCIe/ SATA1 | PCIe | PCIe/ LAN | PCIe/LAN / SATA0 | PCIe/ SATA1 | SATA | SATA | SATA | SATA | N/A | N/A |
| Q170 | PCIe/LAN /SATA0 | PCIe/ SATA1 | PCIe | PCIe/ LAN | PCIe/LAN / SATA0 | PCIe/ SATA1 | PCIe/ SATA | PCIe/ SATA | PCIe/ SATA | PCIe/ SATA | PCIe | PCIe |

***Notes:***
1. Refer to Flexible IO chapter for the additional information.

## Table 1-5. Mobile Client/Server/Mobile SKUs

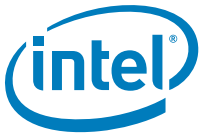
| Features | Mobile |
|---|---|
| | CM236 |
| Intel® Rapid Storage Technology | Full Features[8] |
| Intel® Rapid Storage Technology enterprise | Full Features[8] |
| Total USB 3.0 Ports | Up to 10 |
| Total USB 2.0 Ports | 14[2] |
| Total SATA 3.0 Ports (Max 6 Gb/s) | 8 |
| Total PCI Express* Lanes (Gen) | 20 (3.0) |
| Total PCIe and SATA Express Storage Devices [4,5] | 3[6] |
| Processor dgfx bifurcation support | Yes |

***Notes:***
1. USB 2.0 port numbers: 1-12
2. USB 2.0 port numbers: 1-14
3. Refer to Flexible I/O section for additional information
4. SATA Express Capable Ports (x2)
5. No PCIe SSD Support in Intel® RSTe Driver
6. Intel® RST Support for Workstation only. Intel® RST PCIe supports RAID configuration 0/1/5
7. Intel® RST support for upto 3 PCIe/SATAe devices.
8. Full featured includes SATA RAID 0/1/5/10 support.

## Table 1-6. Mobile Client/Server/Workstation SKUs

| Features | Server | Mobile Client / Server/Workstation |
|---|---|---|
| | C232 | CM236[9] / C236 |
| Intel® Rapid Storage Technology | No | Full Features[8] |
| Intel® Rapid Storage Technology enterprise | Full Features[8] | Full Features[8] |
| Total USB 3.0 Ports | 6 | Up to 10 |
| Total USB 2.0 Ports | 12[1] | 14[2] |
| Total SATA 3.0 Ports (Max 6 Gb/s) | 6 | 8 |
| Total PCI Express* Lanes (Gen) | 8 (3.0) | 20 (3.0) |
| Total PCIe and SATA Express Storage Devices [4,5] | 1 | 3[6] |
| Processor dgfx bifurcation support | Yes | Yes |

***Notes:***
1. USB 2.0 port numbers: 1-12
2. USB 2.0 port numbers: 1-14
3. Refer to Flexible I/O section for additional information
4. SATA Express Capable Ports (x2)
5. No PCIe SSD Support in Intel® RSTe Driver
6. Intel® RST Support for Workstation only. Intel® RST PCIe supports RAID configuration 0/1/5
7. Intel® RST support for up to 3 PCIe/SATAe devices.
8. Full featured includes SATA RAID 0/1/5/10 support.
9. Mobile client CM236 has the same features as Server/WS C236.



### Table 1-7. Mobile Client /Server/Workstation PCH HSIO Detail (Lane 1-13)

| SKU | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CM236 / C236 | USB 3.0/ OTG | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0/ PCIe | USB 3.0/ PCIe | USB 3.0/ PCIe | USB 3.0/ PCIE/ LAN | PCIe/ LAN | PCIe | PCIe |
| C232 | USB 3.0/ OTG | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | USB 3.0 | N/A | N/A | N/A | LAN Only | PCIe/ LAN | PCIe | PCIe |

### Table 1-8. Mobile Client/Server/Workstation PCH HSIO Detail (Lane 14-26)

| SKU | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CM236 / C236 | PCIe | SATA0[1]/ PCIe/LAN | SATA1[1]/ PCIe | PCIe | PCIE/ LAN | SATA0[1]/ PCIE/LAN | SATA1[1]/ PCIe | SATA/ PCIe | SATA/ PCIe | SATA/ PCIe | SATA/ PCIe | SATA/ PCIe | SATA/ PCIe |
| C232 | PCIe | SATA/ PCIe | SATA/ PCIe | PCIe | PCIe/ LAN | SATA | SATA | SATA | SATA | SATA | SATA | N/A | N/A |

***Notes:***
1. Refer to Flexible IO chapter for the additional information.

§ §

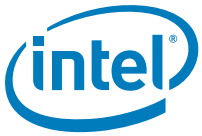
# 2 PCH Controller Device IDs

## 2.1 Device and Revision ID Table

The Revision ID (RID) register is an 8-bit register located at offset 08h in the PCI header of every PCI/PCIe* function. The RID register is used by software to identify a particular component stepping when a driver change or patch unique to that stepping is needed.

**Table 2-1. PCH-H Device and Revision ID Table (Sheet 1 of 2)**

| Device ID (h) | Device Function - Device Description | D1 SRID (h) | Notes |
|---|---|---|---|
| A102 | D23:F0 - SATA Controller (AHCI Mode) | 31 | DT SKUs: Q170, Q150, B150, H170, H110, Z170, CM236. |
| A103 | D23:F0 - SATA Controller (AHCI Mode) | 31 | Mobile H SKUs: HM170, QM170 |
| A106 | D23:F0 - SATA RAID Controller (RAID) | 31 | 3rd Party RAID [AIE=1].<br>DT SKUs: Q170, H170, Z170, CM236. |
| A107 | D23:F0 - SATA Controller (RAID) | 31 | 3rd Party RAID [AIE=1].<br>Mobile H SKUs: HM170, QM170. |
| 2822 | D23:F0 - SATA RAID Controller (RAID) | 31 | Intel® RST RAID [AIE=0, AIES=0].<br>All DT SKUs: Q170, H170, Z170, CM236. |
| 282A | D23:F0 - SATA RAID Controller (RAID) | 31 | Intel® RST RAID [AIE=0, AIES=0].<br>Mobile H SKUs: HM170, QM170. |
| 2826 | D23:F0 - SATA RAID Controller (RAID) | 31 | Intel® RSTe RAID [AIE=0, AIES=1].<br>Server SKUs: CM236.<br>RSTe Device ID for the controller. |
| A110 | D28:F0 - PCI Express* Root Port #1 | F1 | |
| A111 | D28:F1 - PCI Express Root Port #2 | F1 | |
| A112 | D28:F2 - PCI Express Root Port #3 | F1 | |
| A113 | D28:F3 - PCI Express Root Port #4 | F1 | |
| A114 | D28:F4 - PCI Express Root Port #5 | F1 | |
| A115 | D28:F5 - PCI Express Root Port #6 | F1 | |
| A116 | D28:F6 - PCI Express Root Port #7 | F1 | |
| A117 | D28:F7 - PCI Express Root Port #8 | F1 | |
| A118 | D29:F0 - PCI Express Root Port #9 | F1 | |
| A119 | D29:F1 - PCI Express Root Port #10 | F1 | |
| A11A | D29:F2 - PCI Express Root Port #11 | F1 | |
| A11B | D29:F3 - PCI Express Root Port #12 | F1 | |
| A11C | D29:F4 - PCI Express Root Port #13 | F1 | |
| A11D | D29:F5 - PCI Express Root Port #14 | F1 | |
| A11E | D29:F6 - PCI Express Root Port #15 | F1 | |
| A11F | D29:F7 - PCI Express Root Port #16 | F1 | |
| A120 | D31:F1 - P2SB | 31 | |
| A121 | D31:F2 - Power Management Controller | 31 | |



## Table 2-1. PCH-H Device and Revision ID Table (Sheet 2 of 2)

| Device ID (h) | Device Function - Device Description | D1 SRID (h) | Notes |
|---|---|---|---|
| A123 | D31:F4 - SMBus | 31 | |
| A124 | D31:F5 – SPI Controller | 31 | |
| A125 | D31:F6 – GbE Controller | 31 | |
| A126 | D31:F7 - Intel® Trace Hub | 31 | |
| A127 | D30:F0 – UART #0 | 31 | See Note 1 |
| A128 | D30:F1 – UART #1 | 31 | See Note 1 |
| A129 | D30:F2 – GSPI #0 | 31 | See Note 1 |
| A12A | D30:F3 – GSPI #1 | 31 | See Note 1 |
| A12F | D20:F0 – USB 3.0 xHCI Controller | 31 | |
| A130 | D20:F1 – USB Device Controller (OTG) | 31 | |
| A131 | D20:F2 – Thermal Subsystem | 31 | |
| A135 | D19:F0 - ISH | 31 | |
| A13A | D22:F0 - Intel® MEI #1 | 31 | |
| A13B | D22:F1 – Intel® MEI #2 | 31 | |
| A13C | D22:F2 – IDE Redirection | 31 | |
| A13D | D22:F3 – Keyboard and Text (KT) Redirection | 31 | |
| A13E | D22:F4 – Intel® MEI #3 | 31 | |
| A141-A15F | D31:F0 - LPC or eSPI Controller | 31 | **PCH Device IDs:**<br><br>H110: A143<br>H170: A144<br>Z170: A145<br>Q170: A146<br>Q150: A147<br>B150: A148<br>CQM170: A14D<br>HM170: A14E<br>CM236: A150 |
| A160 | D21:F0 – $I^2C$ Controller #0 | 31 | |
| A161 | D21:F1 – $I^2C$ Controller #1 | 31 | |
| A162 | D21:F2 – $I^2C$ Controller #2 | 31 | |
| A163 | D21:F3 – $I^2C$ Controller #3 | 31 | |
| A166 | D25:F0 – UART Controller #2 | 31 | |
| A167 | D27:F0 - PCI Express Root Port #17 | 31 | |
| A168 | D27:F1 - PCI Express Root Port #18 | 31 | |
| A169 | D27:F2 - PCI Express Root Port #19 | 31 | |
| A16A | D27:F3 - PCI Express Root Port #20 | 31 | |
| A170 | D31:F3 - Intel® High Definition Audio (Intel® HD Audio) (Audio, Voice, Speech) | 31 | **PCH Device IDs:**<br><br>Following SKUs are for KBLH platforms:<br>HM175: A171<br>QM175: A171<br>CM238: A171 |
| ***Note:***<br>1. No more than 4 functions in Device 30 can be enabled in PCH. | | | |

# 3 Flexible I/O

## 3.1 Acronyms

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

## 3.2 References

None.

## 3.3 Overview

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

## 3.4 Description

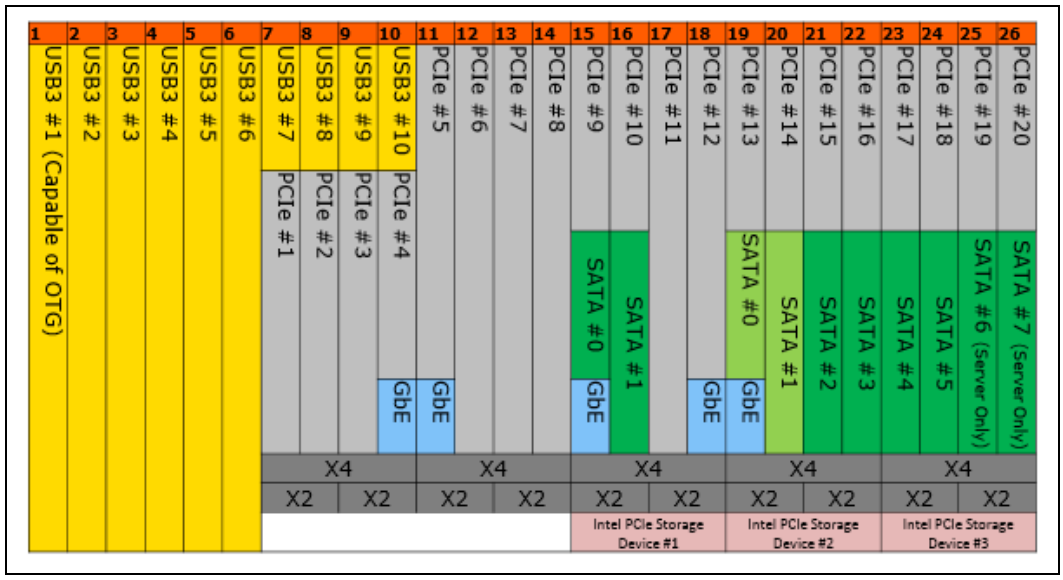
The PCH implements a number of high-speed I/O (HSIO) lanes that are split between the different interfaces, PCIe*, USB 3.0, SATA, GbE, USB Dual Role (OTG). The following figure summarizes the PCH HSIO lanes multiplexing.

The Flexible I/O is configured through soft straps.

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

## 3.4.1 PCH-H Flexible I/O

**Figure 3-1. HSIO Multiplexing on PCH-H**




There are 26 HSIO lanes on the PCH-H, supporting the following port configurations:

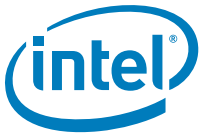
1. Up to 20 PCIe lanes (multiplexed with USB 3.0 ports, SATA Ports)
   - Only a maximum of 16 PCIe ports (or devices) can be enabled at any time.
   - Ports 1-4, Ports 5-8, Ports 9-12, Ports 13-16, and Ports 17-20, can each be individually configured as 4x1, 2x2, 1x2 + 2x1, or 1x4.
2. Up to 6 SATA ports (8 SATA ports for Server Only) (multiplexed with PCIe)
   - SATA Port 0 has the flexibility to be mapped to either PCIe Port 9 or Port 13. Similarly, SATA Port 1 can be mapped to either PCIe Port10 or Port 14.
3. Up to 10 USB 3.0 ports (multiplexed with PCIe)
   - USB Dual Role (OTG) capability is available on USB 3.0 Port 1.
4. One GbE lane
   - GbE can be mapped into one of the PCIe Ports 4-5, Port 9, and Ports 12-13.
   - When GbE is enabled, there can be at most up to 15 PCIe ports enabled.
5. Up to 3 Intel RST for PCIe storage devices supported.
   - Devices can be x2 or x4.
   - Note that the PCIe* storage devices should be implemented on specific PCIe* groups as described in the figure above.
   - Maximum number of devices that can be supported with RST are SKU dependent.

## 3.5 HSIO Port Selection

The HSIO port configuration is statically selected by soft straps.

### 3.5.1 PCIe/SATA Port Selection

In addition to static configuration using soft straps, HSIO lanes that have PCIe/SATA port multiplexing can be configured using SATAXPCIE signaling to support implementation like SATA Express or mSATA, where the port configuration is selected by the type of the add-in card that is used.



# 4 Memory Mapping

## 4.1 Overview

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

## 4.2 Functional Description

### 4.2.1 PCI Devices and Functions

The PCH incorporates a variety of PCI devices and functions, as shown in Table 4-1. If for some reason, the particular system platform does not want to support any one of the Device Functions, with the exception of D30:F0, they can individually be disabled. The integrated Gigabit Ethernet controller will be disabled if no Platform LAN Connect component is detected (See Chapter 21, "Gigabit Ethernet Controller"). When a function is disabled, it does not appear at all to the software. A disabled function will not respond to any register reads or writes, insuring that these devices appear hidden to software.

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

| Device: Functions # | Function Description |
|---|---|
| Bus 0: Device 31: Function 0 | LPC Interface (eSPI Enable Strap = 0)<br>eSPI Interface (eSPI Enable Strap = 1) |
| Bus 0: Device 31: Function 1 | P2SB |
| Bus 0: Device 31: Function 2 | PMC |
| Bus 0: Device 31: Function 3 | Intel® High Definition Audio (Intel® HD Audio) (Audio, Voice, Speech) |
| Bus 0: Device 31: Function 4 | SMBus Controller |
| Bus 0: Device 31: Function 5 | SPI |
| Bus 0: Device 31: Function 6 | GbE Controller |
| Bus 0: Device 31: Function 7 | Intel® Trace Hub |
| Bus 0: Device 30: Function 0 | UART #0 |
| Bus 0: Device 30: Function 1 | UART #1 |
| Bus 0: Device 30: Function 2 | SPI #0 |
| Bus 0: Device 29: Function 0 | PCI Express Port 9 |
| Bus 0: Device 29: Function 1 | PCI Express Port 10 |
| Bus 0: Device 29: Function 2 | PCI Express Port 11 |
| Bus 0: Device 29: Function 3 | PCI Express Port 12 |
| Bus 0: Device 29: Function 4 | PCI Express Port 13 |
| Bus 0: Device 29: Function 5 | PCI Express Port 14 |
| Bus 0: Device 29: Function 6 | PCI Express Port 15 |
| Bus 0: Device 29: Function 7 | PCI Express Port 16 |
| Bus 0: Device 28: Function 0 | PCI Express Port 1 |

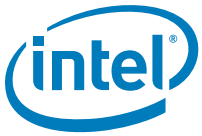
**Table 4-1. PCI Devices and Functions (Sheet 2 of 2)**

| Device: Functions # | Function Description |
|---|---|
| Bus 0: Device 28: Function 1 | PCI Express Port 2 |
| Bus 0: Device 28: Function 2 | PCI Express Port 3 |
| Bus 0: Device 28: Function 3 | PCI Express Port 4 |
| Bus 0: Device 28: Function 4 | PCI Express Port 5 |
| Bus 0: Device 28: Function 5 | PCI Express Port 6 |
| Bus 0: Device 28: Function 6 | PCI Express Port 7 |
| Bus 0: Device 28: Function 7 | PCI Express Port 8 |
| Bus 0: Device 27: Function 0 | PCI Express Port 17 |
| Bus 0: Device 27: Function 1 | PCI Express Port 18 |
| Bus 0: Device 27: Function 2 | PCI Express Port 19 |
| Bus 0: Device 27: Function 3 | PCI Express Port 20 |
| Bus 0: Device 25: Function 0 | UART Controller #2 |
| Bus 0: Device 25: Function 1 | $I^2C$ Controller #5 |
| Bus 0: Device 25: Function 2 | $I^2C$ Controller #4 |
| Bus 0: Device 23: Function 0 | SATA Controller |
| Bus 0: Device 22: Function 0 | Intel® MEI #1 |
| Bus 0: Device 22: Function 1 | Intel® MEI # |
| Bus 0: Device 22: Function 2 | IDE Redirection (IDE-R) |
| Bus 0: Device 22: Function 3 | Keyboard and Text (KT) Redirection |
| Bus 0: Device 22: Function 4 | Intel® MEI #3 |
| Bus 0: Device 21: Function 0 | $I^2C$ Controller #0 |
| Bus 0: Device 21: Function 1 | $I^2C$ Controller #1 |
| Bus 0: Device 21: Function 2 | $I^2C$ Controller #2 |
| Bus 0: Device 21: Function 3 | $I^2C$ Controller #3 |
| Bus 0: Device 20: Function 0 | USB 3.0 xHCI Controller |
| Bus 0: Device 20: Function 1 | USB Device Controller (OTG) |
| Bus 0: Device 20: Function 2 | Thermal Subsystem |
| Bus 0: Device 19: Function 0 | Integrated Sensor Hub |
| ***Note:*** When a device or function is disabled, it is not reported to the software and will not respond to any register reads or writes. | |

## 4.2.2 Fixed I/O Address Ranges

Table 4-2, "Fixed I/O Ranges Decoded by PCH" shows the Fixed I/O decode ranges from the processor perspective. Note that for each I/O range, there may be separate behavior for reads and writes. DMI cycles that go to target ranges that are marked as Reserved will be handled by the PCH; writes are ignored and reads will return all 1s.

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



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

| I/O Address | Read Target | Write Target | Internal Unit | Enable/Disable |
|---|---|---|---|---|
| 20h – 21h | Interrupt Controller | Interrupt Controller | Interrupt | None |
| 24h – 25h | Interrupt Controller | Interrupt Controller | Interrupt | None |
| 28h – 29h | Interrupt Controller | Interrupt Controller | Interrupt | None |
| 2Ch – 2Dh | Interrupt Controller | Interrupt Controller | Interrupt | None |
| 2Eh – 2Fh | LPC/eSPI | LPC/eSPI | Forwarded to LPC/eSPI | Yes<br>IOE.SE |
| 30h – 31h | Interrupt Controller | Interrupt Controller | Interrupt | None |
| 34h – 35h | Interrupt Controller | Interrupt Controller | Interrupt | None |
| 38h – 39h | Interrupt Controller | Interrupt Controller | Interrupt | None |
| 3Ch – 3Dh | Interrupt Controller | Interrupt Controller | Interrupt | None |
| 40h | Timer/Counter | Timer/Counter | 8254 Timer | None |
| 42h – 43h | Timer/Counter | Timer/Counter | 8254 Timer | None |
| 4Eh – 4Fh | LPC/eSPI | LPC/eSPI | Forwarded to LPC/eSPI | Yes<br>IOE.ME2 |
| 50h | Timer/Counter | Timer/Counter | 8254 Timer | None |
| 52h – 53h | Timer/Counter | Timer/Counter | 8254 Timer | None |
| 60h | LPC/eSPI | LPC/eSPI | Forwarded to LPC/eSPI | Yes w/ 60h<br>IOE.KE |
| 61h | NMI Controller | NMI Controller | Processor I/F | None |
| 62h | Microcontroller | Microcontroller | Forwarded to LPC/eSPI | Yes w/ 66h<br>IOE.ME1 |
| 63h | NMI Controller[1] | NMI Controller[1] | Processor I/F | Yes, alias to 61h<br>GCS.P61AE |
| 64h | Microcontroller | Microcontroller | Forwarded to LPC/eSPI | Yes w/ 60h and<br>IOE.KE |
| 65h | NMI Controller[1] | NMI Controller[1] | Processor I/F | Yes, alias to 61h<br>GCS.P61AE |
| 66h | Microcontroller | Microcontroller | Forwarded to LPC/eSPI | Yes w/ 62h<br>IOE.ME1 |
| 67h | NMI Controller[1] | NMI Controller[1] | Processor I/F | Yes, alias to 61h<br>GCS.P61AE |
| 70h | RTC Controller | NMI and RTC Controller | RTC | None |
| 71h | RTC Controller | RTC Controller | RTC | None |
| 72h | RTC Controller | RTC Controller | RTC | Yes, w/ 72h<br>RC.UE |
| 73h | RTC Controller | RTC Controller | RTC | Yes, w/ 73h<br>RC.UE |
| 74h | RTC Controller | RTC Controller | RTC | None |
| 75h | RTC Controller | RTC Controller | RTC | None |

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

| I/O Address | Read Target | Write Target | Internal Unit | Enable/Disable |
|---|---|---|---|---|
| 76h – 77h | RTC Controller | RTC Controller | RTC | Yes<br>RC.UE |
| 80h | LPC/eSPI or PCIe | LPC/eSPI or PCIe | LPC/eSPI or PCIe | GCS.RPR |
| 84h – 86h | Reserved | LPC/eSPI or PCIe | LPC/eSPI or PCIe | GCS.RPR |
| 88h | Reserved | LPC/eSPI or PCIe | LPC/eSPI or PCIe | GCS.RPR |
| 8Ch – 8Eh | Reserved | LPC/eSPI or PCIe | LPC/eSPI or PCIe | GCS.RPR |
| 90h | (Alias to 80h) | (Alias to 80h) | Forwarded to LPC/eSPI | Yes, alias to 80h |
| 92h | Reset Generator | Reset Generator | Processor I/F | None |
| 94h – 96h | (Aliases to 8xh) | (Aliases to 8xh) | Forwarded to LPC/eSPI | Yes, aliases to 8xh |
| 98h | (Alias to 88h) | (Alias to 88h) | Forwarded to LPC/eSPI | Yes, alias to 88h |
| 9Ch – 9Eh | (Alias to 8xh) | (Aliases to 8xh) | Forwarded to LPC/eSPI | Yes, aliases to 8xh |
| A0h – A1h | Interrupt Controller | Interrupt Controller | Interrupt | None |
| A4h – A5h | Interrupt Controller | Interrupt Controller | Interrupt | None |
| A8h – A9h | Interrupt Controller | Interrupt Controller | Interrupt | None |
| ACh – ADh | Interrupt Controller | Interrupt Controller | Interrupt | None |
| B0h – B1h | Interrupt Controller | Interrupt Controller | Interrupt | None |
| B2h – B3h | Power Management | Power Management | Power Management | None |
| B4h – B5h | Interrupt Controller | Interrupt Controller | Interrupt | None |
| B8h – B9h | Interrupt Controller | Interrupt Controller | Interrupt | None |
| BCh – BDh | Interrupt Controller | Interrupt Controller | Interrupt | None |
| 200 – 207h | Gameport Low | Gameport Low | Forwarded to LPC/eSPI | Yes<br>IOE.LGE |
| 208–20Fh | Gameport High | Gameport High | Forwarded to LPC/eSPI | Yes<br>IOE.HGE |
| 4D0h – 4D1h | Interrupt Controller | Interrupt Controller | Interrupt Controller | None |
| CF9h | Reset Generator | Reset Generator | Interrupt controller | None |

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

## 4.2.3 Variable I/O Decode Ranges

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

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

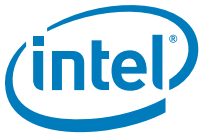


**Table 4-3. Variable I/O Decode Ranges**

| Range Name | Mappable | Size (Bytes) | Target |
|---|---|---|---|
| ACPI | Anywhere in 64K I/O Space | 96 | Power Management |
| 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 Peripheral |
| Serial Port 1 | 8 Ranges in 64K I/O Space | 8 | LPC Peripheral |
| Serial Port 2 | 8 Ranges in 64K I/O Space | 8 | LPC Peripheral |
| Floppy Disk Controller | 2 Ranges in 64K I/O Space | 8 | LPC Peripheral |
| LPC Generic 1 | Anywhere in 64K I/O Space | 4 to 256 bytes | LPC/eSPI |
| LPC Generic 2 | Anywhere in 64K I/O Space | 4 to 256 bytes | LPC/eSPI |
| LPC Generic 3 | Anywhere in 64K I/O Space | 4 to 256 bytes | LPC/eSPI |
| LPC Generic 4 | Anywhere in 64K I/O Space | 4 to 256 bytes | LPC/eSPI |
| I/O Trapping Ranges | Anywhere in 64K I/O Space | 1 to 256 bytes | Trap |
| Serial ATA Index/Data Pair | Anywhere in 64K I/O Space | 16 | SATA Host Controller |
| PCI Express* Root Ports | Anywhere in 64K I/O Space | I/O Base/Limit | PCI Express Root Ports 1-12 |
| Keyboard and Text (KT) | Anywhere in 64K I/O Space | 8 | Intel® AMT Keyboard and Text Redirection |
| ***Note:*** All ranges are decoded directly from DMI. | | | |

# 4.3 Memory Map

Table 4-4, "PCH Memory Decode Ranges (Processor Perspective)" shows (from the Processor perspective) the memory ranges that the PCH will decode. Cycles that arrive from DMI that are not directed to any of the internal memory targets that decode directly from DMI will be master aborted.

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

***Note:*** Total ports are different for the different SKUs.

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

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

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

| Memory Range | Target | Dependency/Comments |
|---|---|---|
| FEC30000 – FEC37FFF | PCIe* port 5 | PCIe* root port 5 APIC Enable (PAE) set |
| FEC38000 – FEC3FFFF | PCIe* port 6 | PCIe* root port 6 APIC Enable (PAE) set |
| FEC40000 – FEC47FFF | PCIe* port 7 | PCIe* root port 7 APIC Enable (PAE) set |
| FEC48000 – FEC4FFFF | PCIe* port 8 | PCIe* root port 8 APIC Enable (PAE) set |
| FEC50000 – FEC57FFF | PCIe* port 9 | PCIe* root port 9 APIC Enable (PAE) set |
| FEC58000 – FEC5FFFF | PCIe* port 10 | PCIe* root port 10 APIC Enable (PAE) set |
| FEC70000 - FEC77FFF | PCIe port 13 | PCIe root port 13 APIC Enable (PAE) set |
| FEC78000 - FEC7FFFF | PCIe port 14 | PCIe root port 14 APIC Enable (PAE) set |
| FEC80000 - FEC87FFF | PCIe port 15 | PCIe root port 15 APIC Enable (PAE) set |
| FEC88000 - FEC8FFFF | PCIe port 16 | PCIe root port 16 APIC Enable (PAE) set |
| FEC90000 - FEC97FFF | PCIe port 17 | PCIe root port 17 APIC Enable (PAE) set |
| FEC98000 - FEC9FFFF | PCIe port 18 | PCIe root port 18 APIC Enable (PAE) set |
| FECA0000 - FECA7FFF | PCIe port 19 | PCIe root port 19 APIC Enable (PAE) set |
| FECA8000 - FECAFFFF | PCIe port 20 | PCIe root port 20 APIC Enable (PAE) set |
| FFC0 0000 – FFC7 FFFF<br>FF80 0000 – FF87 FFFF | LPC/eSPI or SPI | Bit 8 in BIOS Decode Enable Register |
| FFC8 0000 – FFCF FFFF<br>FF88 0000 – FF8F FFFF | LPC/eSPI or SPI | Bit 9 in BIOS Decode Enable Register |
| FFD0 0000 – FFD7 FFFF<br>FF90 0000 – FF97 FFFF | LPC/eSPI or SPI | Bit 10 in BIOS Decode Enable Register is set |
| FFD8 0000 – FFDF FFFF<br>FF98 0000 – FF9F FFFF | LPC/eSPI or SPI | Bit 11 in BIOS Decode Enable Register is set |
| FFE0 000 – FFE7 FFFF<br>FFA0 0000 – FFA7 FFFF | LPC/eSPI or SPI | Bit 12 in BIOS Decode Enable Register is set |
| FFE8 0000 – FFEF FFFF<br>FFA8 0000 – FFAF FFFF | LPC/eSPI or SPI | Bit 13 in BIOS Decode Enable Register is set |
| FFF0 0000 – FFF7 FFFF<br>FFB0 0000 – FFB7 FFFF | LPC/eSPI or SPI | Bit 14 in BIOS Decode Enable Register is set |
| FFF8 0000 – FFFF FFFF<br>FFB8 0000 – FFBF FFFF | LPC/eSPI or SPI | Always enabled.<br>The top two 64-KB blocks in this range can be swapped by the PCH. |
| FF70 0000 – FF7F FFFF<br>FF30 0000 – FF3F FFFF | LPC/eSPI or SPI | Bit 3 in BIOS Decode Enable Register is set |
| FF60 0000 – FF6F FFFF<br>FF20 0000 – FF2F FFFF | LPC/eSPI or SPI | Bit 2 in BIOS Decode Enable Register is set |
| FF50 0000 – FF5F FFFF<br>FF10 0000 – FF1F FFFF | LPC/eSPI or SPI | Bit 1 in BIOS Decode Enable Register is set |
| FF40 0000 – FF4F FFFF<br>FF00 0000 – FF0F FFFF | LPC/eSPI or SPI | Bit 0 in BIOS Decode Enable Register is set |
| FED0 X000h – FED0 X3FFh | HPET | BIOS determines “fixed” location which is one of four 1-KB ranges where X (in the first column) is 0h, 1h, 2h, or 3h |
| FED4_0000h – FED4_7FFFh | LPC or SPI (set by strap) | TPM and Trusted Mobile KBC |
| FED5_0000h – FED5_FFFFh | Intel® ME | Always enabled |
| 64 KB anywhere in 64-bit address range | USB 3.0 Host Controller | Enable via standard PCI mechanism (Device 20, Function 0) |

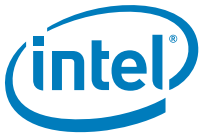


**Table 4-4. PCH Memory Decode Ranges (Processor Perspective) (Sheet 3 of 3)**

| Memory Range | Target | Dependency/Comments |
|---|---|---|
| 2 MB anywhere in 4-Gb range | OTG | Enable via standard PCI mechanism (Device 20, Function 1) |
| 24 KB anywhere in 4-Gb range | OTG | Enable via standard PCI mechanism (Device 20, Function 1) |
| 16 KB anywhere in 64-bit addressing space | Intel® HD Audio Subsystem | Enable via standard PCI mechanism (Device 31, Function 3) |
| 4 KB anywhere in 64-bit addressing space | Intel® HD Audio Subsystem | Enable via standard PCI mechanism (Device 31, Function 3) |
| 64 KB anywhere in 64-bit addressing space | Intel® HD Audio Subsystem | Enable via standard PCI mechanism (Device 31, Function 3) |
| 64 KB anywhere in 4-GB range | LPC/eSPI | LPC Generic Memory Range. Enable via setting bit[0] of the LPC Generic Memory Range register (D31:F0:offset 98h) |
| 32 bytes anywhere in 64-bit address range | SMBus | Enable via standard PCI mechanism (Device 31: Function 4) |
| 2 KB anywhere above 64-KB to 4-GB range | SATA Host Controller | AHCI memory-mapped registers. Enable via standard PCI mechanism (Device 23: Function 0) |
| Memory Base/Limit anywhere in 4-GB range | PCI Express Root Ports 1-20 | Enable via standard PCI mechanism |
| Prefetchable Memory Base/ Limit anywhere in 64-bit address range | PCI Express Root Ports 1-20 | Enable via standard PCI mechanism |
| 4 KB anywhere in 64-bit address range | Thermal Reporting | Enable via standard PCI mechanism (Device 20: Function 2) |
| 16 bytes anywhere in 64-bit address range | Intel® MEI#1, #2, #3, | Enable via standard PCI mechanism (Device 22: Function 0-1, 4) |
| 4 KB anywhere in 4-GB range | Intel® AMT Keyboard and Text Redirection | Enable via standard PCI mechanism (Device 22: Function 3) |
| Twelve 4-KB slots anywhere in 64-bit address range | Intel Serial Interface controllers | Enable via standard PCI mechanism (Device 30: Function[7:0], Device 21: Function [6:0] |
| 1 MB (BAR0) or 4 KB (BAR1) in 4-GB range | Integrated Sensor Hub | Enable via standard PCI mechanism (Device 19: Function 0) |

## 4.3.1 Boot Block Update Scheme

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

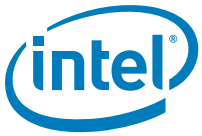
For FHW when top swap is enabled, accesses to FFFF_0000h-FFFF_FFFFh are directed to FFFE_0000h-FFFE_FFFFh and vice versa. When the Top Swap Enable bit is 0, the PCH will not invert A16.

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

### Table 4-5. SPI Mode Address Swapping

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

§ §



# 5 System Management

## 5.1 Acronyms

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

## 5.2 References

None.

## 5.3 Overview

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

## 5.4 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
- Second hard-coded timer timeout to generate reboot:
  - This second timer is used only after the 1st timeout occurs
  - The second timeout allows for automatic system reset and reboot if a HW error is detected
  - Option to prevent reset the second timeout via HW strap
- Processor present detection:
  - Detects if processor fails to fetch the first instruction after reset
- Various Error detection (such as ECC Errors) indicated by host controller:
  - Can generate SMI#, SCI, SERR, NMI, or TCO interrupt
- Intruder Detect input:
  - Can generate TCO interrupt or SMI# when the system cover is removed
  - INTRUDER# allowed to go active in any power state, including G3
- Detection of bad BIOS Flash programming:
  - Detects if data on first read is FFh (indicates that BIOS flash is not programmed)

## 5.4.1 Theory of Operation

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

### 5.4.1.1 Detecting a System Lockup

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

### 5.4.1.2 Handling an Intruder

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

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

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

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

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

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

### 5.4.1.3 Detecting Improper Flash Programming

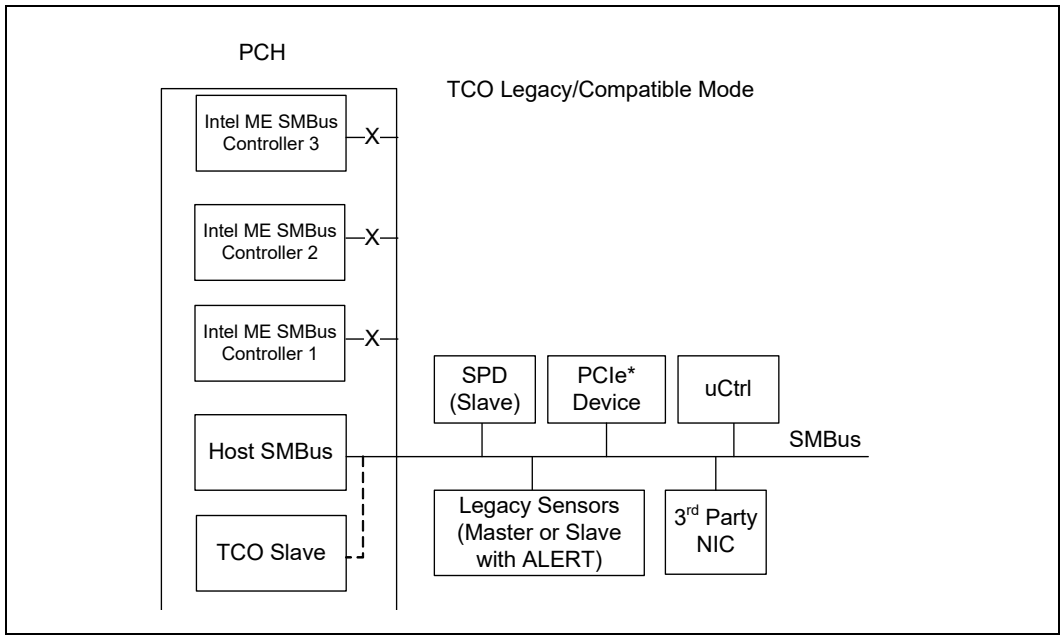
The PCH can detect the case where the BIOS flash is not programmed. This results in the first instruction fetched to have a value of FFh. If this occurs, the PCH sets the BAD_BIOS bit.

## 5.4.2 TCO Modes

### 5.4.2.1 TCO Compatible Mode

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

**Figure 5-1. TCO Compatible Mode SMBus Configuration**




In TCO Legacy/Compatible mode the PCH can function directly with an external LAN controller or equivalent external LAN controller to report messages to a network management console without the aid of the system processor. This is crucial in cases where the processor is malfunctioning or cannot function due to being in a low-power state. Table 5-1 includes a list of events that will report messages to the network management console.

**Table 5-1. Event Transitions that Cause Messages**

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

### 5.4.2.2 Advanced TCO Mode

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

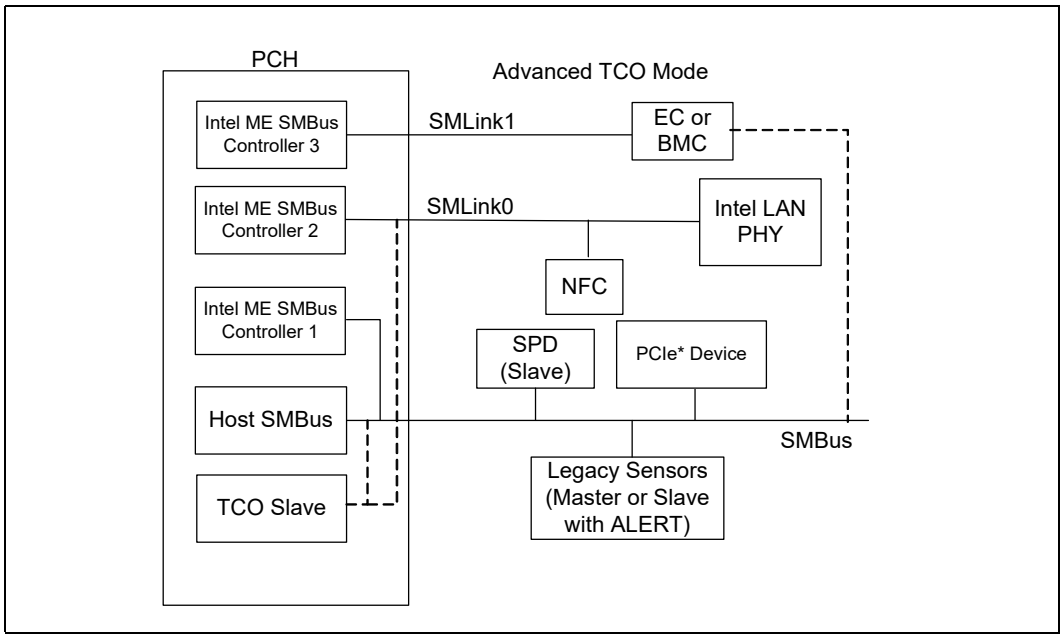
In this mode, the Intel® ME SMBus controllers must be enabled by soft strap in the flash descriptor. See Figure 5-2 for more details.

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

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

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

**Figure 5-2. Advanced TCO Mode**




§ §

# 6 High Precision Event Timer (HPET)

## 6.1 References

None.

## 6.2 Overview

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

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

TImer 0 supports periodic interrupts.

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

### 6.2.1 Timer Accuracy

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

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

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

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

### 6.2.2 Timer Off-load

The PCH supports a timer off-load feature that allows the HPET timers to remain operational during very low power S0 operational modes when the 24-MHz clock is disabled. The clock source during this off-load is the Real Time Clock's 32.768-KHz clock. This clock is calibrated against the 24-MHz clock during boot time to an accuracy that ensures the error introduced by this off-load is less than 10 ppb (.000001%).

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

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.

### 6.2.3 Off-loadable Timer

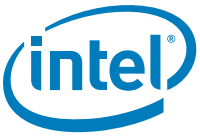
The Off-loadable Timer Block consists of a 64b fast clock counter and an 82b slow clock counter. During fast clock mode the counter increments by one on every rising edge of the fast clock. During slow clock mode, the 82-bit slow clock counter will increment by the value provided by the Off-load Calibrator.

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

When transitioning from fast clock to slow clock, the fast clock value will be loaded into the upper 64b of the 82b counter, with the 18 LSBs set to zero. The actual transition 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.



## 6.2.4 Interrupt Mapping

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

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

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

**Table 6-1. Legacy Replacement Routing**

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

### 6.2.4.2 Mapping Option #2 (Standard Option)

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

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

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

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

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

***Note:*** Interrupts from Timer 4, 5, 6, 7 can only be delivered using processor message interrupts.

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

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

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

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

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

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

## 6.2.5 Periodic Versus Non-Periodic Modes

### 6.2.5.1 Non-Periodic Mode

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

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

All of the timers support non-periodic mode.

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

### 6.2.5.2 Periodic Mode

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

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

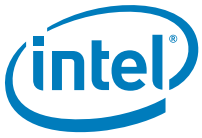
1. Software clears the ENABLE_CNF bit to prevent any interrupts.
2. Software Clears the main counter by writing a value of 00h to it.
3. Software sets the TIMER0_VAL_SET_CNF bit.
4. Software writes the new value in the TIMER0_COMPARATOR_VAL register.
5. Software sets the ENABLE_CNF bit to enable interrupts.

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

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

## 6.2.6 Enabling the Timers

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



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

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

### 6.2.7 Interrupt Levels

Interrupts directed to the internal 8259s are active high. See Section 22.8, "Advanced Programmable Interrupt Controller (APIC) (D31:F0)" for information regarding the polarity programming of the I/O APIC for detecting internal interrupts.

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

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

### 6.2.8 Handling Interrupts

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

### 6.2.9 Issues Related to 64-Bit Timers with 32-Bit Processors

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

§ §

# 7 Thermal Management

## 7.1 PCH Thermal Sensor

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

### 7.1.1 Modes of Operation

The DTS has two usages when enabled:

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

### 7.1.2 Temperature Trip Point

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

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

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

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

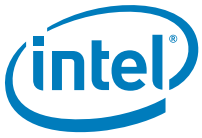
### 7.1.3 Thermal Sensor Accuracy ($T_{accuracy}$)

The PCH thermal sensor accuracy is:

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

### 7.1.4 Thermal Reporting to an EC

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



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

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

### 7.1.5 Thermal Trip Signal (PCHHOT#)

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

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

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

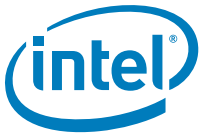
§ §

# 8 Power and Ground Signals

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

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

| Name | Description |
|---|---|
| **VCCPRIM_1p0** | **Primary Well 1.0 V:** For I/O blocks, core logic, SRAM, USB AFE Digital Logic, Processor sideband signals, JTAG, and Thermal Sensor. |
| **VCCMPHY_1p0** | **Mod-PHY Primary 1.0 V**: Primary supply for PCIe/DMI/USB 3.0/SATA/MIPI M-PHY* logic |
| **VCCAPLLEBB_1p0** | **PCIe PLL EBB Primary 1.0 V:** EBB contains primary supply for PCIe PLL dividers and lane drivers. |
| **VCCAMPHYPLL_1p0** | **Analog supply for USB 3.0, PCIe Gen 2, SATA and PCIe Gen 3 PLL Primary 1.0V:** Filtering is required. |
| **VCCMIPIPLL_1p0** | **Analog supply for MIPI* PLL Primary 1.0V:** Filtering is required. |
| **VCCUSB2PLL_1p0** | **Analog supply for USB 2.0 PLL for VRM Primary 1.0V:** Filtering is required. |
| **VCCHDAPLL_1p0** | **Analog supply for Audio PLL for VRM Primary 1.0V:** Filtering is required. |
| **VCCCLK1, VCCCLK2, VCCCLK3, VCCCLK4, VCCCLK5** | **Clock Buffers Primary 1.0 V:** Filtering is required. |
| **VCCPGPPA** | **Group A Primary Well GPIOs 3.3 V or 1.8 V** |
| **VCCPGPPBCH** | **Group B, C and H Primary Well GPIOs 3.3 V or 1.8 V** |
| **VCCPGPPD** | **Group D Primary Well GPIOs 3.3 V or 1.8 V** |
| **VCCPGPPEF** | **Group E and F Primary Well GPIOs 3.3 V or 1.8 V** |
| **VCCPGPPG** | **Group G Primary Well GPIOs 3.3 V or 1.8 V** |
| **VCCATS** | **Thermal Sensor CORE Well 3.3 V** This rail must be connected to an S0 only supply and must be off in Sx states. |
| **VCCHDA** | **Intel® HD Audio Power 3.3 V, 1.8 V or 1.5 V.** For Intel® High Definition Audio. |
| **VCCSPI** | **SPI Primary Well 3.3 V or 1.8 V** |
| **VCCPRIM_3p3** | **Primary Well 3.3 V.** This rail supplies power for High Voltage CMOS, including display and Group I GPIOs. |
| **VCCRTCPRIM_3p3** | **RTC Logic Primary Well 3.3 V.** This power supplies the RTC internal VRM. It will be off during Deep Sx mode. |
| **DCPDSW_1p0** | **Deep Sx Well 1.0 V.** This rail is generated by on die DSW voltage regulator to supply DSW GPIOs, DSW core logic and DSW USB 2.0 logic. Board needs to connect 1 uF capacitor to this rail and power should NOT be driven from the board. When primary well power is up, this rail is bypassed from VCCPRIM_1p0. |
| **VCCDSW_3p3** | **Deep Sx Well for GPD GPIOs and USB 2.0** |
| **DCPRTC** | **RTC de-coupling capacitor only.** This rail should NOT be driven. |



| Name | Description |
|---|---|
| **VCCRTC** | **RTC Well Supply.** This rail can drop to 2.0 V if all other planes are off. This power is not expected to be shut off unless the RTC battery is removed or drained.<br><br>***Note:*** VCCRTC nominal voltage is 3.0V. This rail is intended to always come up first and always stay on. It should NOT be power cycled regularly on non-coin battery designs.<br>***Note:*** Implementation should not attempt to clear CMOS by using a jumper to pull VCCRTC low. Clearing CMOS can be done by using a jumper on RTCRST# or GPI. |
| **VSS** | **Ground** |

§ §

# 9 Pin Straps

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

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

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

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

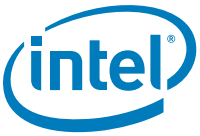
## Table 9-1. Functional Strap Definitions (Sheet 2 of 3)

| Signal | Usage | When Sampled | Comment |
|---|---|---|---|
| **GSPI1_MOSI/ GPP_B22** | Boot BIOS Strap Bit BBS | Rising edge of PCH_PWROK | This Signal has a weak internal Pull-down.<br>This field determines the destination of accesses to the BIOS memory range. Also controllable using Boot BIOS Destination bit (Bus0, Device31, Function0, offset BCh, bit 6).<br>**Bit 6** — **Boot BIOS Destination**<br>0 — SPI (Default)<br>1 — LPC<br>***Notes:***<br>1. The internal Pull-down is disabled after PLTRST# de-asserts.<br>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.<br>3. Boot BIOS Destination select to LPC by functional strap or using Boot BIOS Destination bit will not affect SPI accesses initiated by Intel ME or Integrated GbE LAN.<br>4. This signal is in the primary well. |
| **SML0ALERT#/ GPP_C5** | eSPI or LPC | Rising edge of RSMRST# | This signal has a weak internal Pull-down.<br>0 = **LPC** Is selected for EC. (Default)<br>1 = **eSPI** Is selected for EC.<br>***Notes:***<br>1. The internal Pull-down is disabled after RSMRST# de-asserts.<br>2. This signal is in the primary well. |
| **SPI0_MOSI** | Reserved | Rising edge of RSMRST# | This signal has an internal Pull-up.<br>This strap should sample HIGH. There should NOT be any on-board devices driving it to opposite direction during strap sampling. |
| **SPI0_MISO** | Reserved | Rising edge of RSMRST# | This signal has an internal Pull-up.<br>This strap should sample HIGH. There should NOT be any on-board devices driving it to opposite direction during strap sampling. |
| **SML1ALERT#/ PCHHOT#/ GPP_B23** | Reserved | Rising edge of RSMRST# | This signal has an internal Pull-down.<br>This strap should sample LOW. There should NOT be any on-board devices driving it to opposite direction during strap sampling.<br>***Note:*** When used as PCHHOT#, a 150k weak board Pull-up is recommended to ensure it does not override the internal Pull-down strap sampling. |
| **SPI0_IO2** | Reserved | Rising edge of RSMRST# | This signal has an internal Pull-up.<br>This strap should sample HIGH. There should NOT be any on-board devices driving it to opposite direction during strap sampling. |
| **SPI0_IO3** | Reserved | Rising edge of RSMRST# | This signal has an internal Pull-up.<br>This strap should sample HIGH. There should NOT be any on-board devices driving it to opposite direction during strap sampling. |

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

| Signal | Usage | When Sampled | Comment |
|---|---|---|---|
| **HDA_SDO** | Flash Descriptor Security Override | Rising edge of PCH_PWROK | This signal has a weak internal Pull-down.<br>0 = **Enable** security measures defined in the Flash Descriptor. (Default)<br>1 = **Disable** Flash Descriptor Security (override). This strap should only be asserted high using external Pull-up in manufacturing/debug environments ONLY.<br>***Notes:***<br>1. The internal Pull-down is disabled after PLTRST# de-asserts.<br>2. Asserting HDA_SDO high on the rising edge of PCH_PWROK will also halt Intel Management Engine after Chipset bring up and disable runtime Intel ME features. This is a debug mode and must not be asserted after manufacturing/debug.<br>3. This signal is in the primary well. |
| **DDPB_CTRLDATA/ GPP_I6** | Display Port B Detected | Rising edge of PCH_PWROK | This signal has a weak internal Pull-down.<br>0 = Port B is not detected. (Default)<br>1 = Port B is detected.<br>***Notes:***<br>1. The internal Pull-down is disabled after PLTRST# de-asserts.<br>2. This signal is in the primary well. |
| **DDPC_CTRLDATA/ GPP_I8** | Display Port C Detected | Rising edge of PCH_PWROK | This signal has a weak internal Pull-down.<br>0 = Port C is not detected. (Default)<br>1 = Port C is detected.<br>***Notes:***<br>1. The internal Pull-down is disabled after PLTRST# de-asserts.<br>2. This signal is in the primary well. |
| **DDPD_CTRLDATA / GPP_I10** | Display Port D Detected | Rising edge of PCH_PWROK | This signal has a weak internal pull-down.<br>0 = Port D is not detected. (Default)<br>1 = Port D is detected.<br>***Notes:***<br>1. The internal pull-down is disabled after PLTRST# de-asserts.<br>2. This signal is in the primary well. |
| **GPP_H12** | Reserved | Rising edge of RSMRST# | This signal has a weak internal pull-down.<br>This strap should sample LOW. There should NOT be any on-board devices driving it to opposite direction during strap sampling.<br>***Note:*** The pull-down resistor is disabled after RSMRST# de-asserts |

§ §



# 10 Electrical Characteristics

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

## 10.1 Absolute Maximum Ratings

**Table 10-1. PCH Absolute Maximum Ratings**

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

Table 10-1 specifies absolute maximum and minimum ratings. At conditions outside functional operation condition limits, but within absolute maximum and minimum ratings, neither functionality nor long-term reliability can be expected. If a device is returned to conditions within functional operation limits after having been subjected to conditions outside these limits (but within the absolute maximum and minimum ratings) the device may be functional, but with its lifetime degraded depending on exposure to conditions exceeding the functional operation condition limits.

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

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

## 10.2 Thermal Specification

**Table 10-2. Thermal Design Power**

| SkU | Thermal Design Power (TDP) |
|---|---|
| Desktop: H110, H170, Z170, B150, Q150, Q170<br>Server: C232, C236 | 6.0 Watts |
| Mobile: CM236 | 3.67 Watts |
| Mobile: HM170, QM170 | 2.6 Watts |

## 10.3 PCH Power Supply Range

**Table 10-3. PCH Power Supply Range**

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

## 10.4 General DC Characteristics

**Table 10-4. PCH-H Measured $I_{cc}$ (Desktop and Server SkUs) (Sheet 1 of 2)**

| Voltage Rail | Voltage (V) | S0 Iccmax Current[3] (A) | Sx Icc Idle Current[6] (mA) | Deep Sx Icc Idle Current (mA) | G3 (µA) |
|---|---|---|---|---|---|
| VCCPRIM_1p0 | 1.0 | 6.010 | 87.40 | 0 | 0 |
| VCCCLK1 | 1.0 | 0.035 | 0.194 | 0 | 0 |
| VCCCLK2 | 1.0 | 0.204 | 0.645 | 0 | 0 |
| VCCCLK3 | 1.0 | 0.057 | 0.220 | 0 | 0 |
| VCCCLK4 | 1.0 | 0.036 | 0.363 | 0 | 0 |
| VCCCLK5 | 1.0 | 0.010 | 1.380 | 0 | 0 |
| VCCMPHY_1p0 | 1.0 | See Table 10-6 | 4.00 | 0 | 0 |
| VCCHDAPLL_1p0 | 1.0 | 0.033 | 0.481 | 0 | 0 |
| VCCAMPHYPLL_1p0 | 1.0 | 0.080 | 0.550 | 0 | 0 |
| VCCAPLLEBB_1p0 | 1.0 | 0.075 | 0.150 | 0 | 0 |
| VCCMIPIPLL_1p0 | 1.0 | 0.036 | 0.200 | 0 | 0 |
| VCCUSB2PLL_1p0 | 1.0 | 0.012 | 0.983 | 0 | 0 |
| VCCPGPPA | 3.3 | 0.082 | 1.470 | 0 | 0 |
| | 1.8 | 0.082[8] | 1.470[8] | 0 | 0 |
| VCCPGPPBCH | 3.3 | 0.229 | 0.920 | 0 | 0 |
| | 1.8 | 0.229[8] | 0.920[8] | 0 | 0 |
| VCCPGPPD | 3.3 | 0.078 | 0.930 | 0 | 0 |
| | 1.8 | 0.078[8] | 0.930[8] | 0 | 0 |
| VCCPGPPEF | 3.3 | 0.114 | 0.600 | 0 | 0 |
| | 1.8 | 0.114[8] | 0.600[8] | 0 | 0 |
| VCCPGPPG | 3.3 | 0.065 | 0.624 | 0 | 0 |
| | 1.8 | 0.065[8] | 0.624[8] | 0 | 0 |
| VCCSPI | 3.3 | 0.029 | 0.432 | 0 | 0 |
| | 1.8 | 0.029[8] | 0.432[8] | 0 | 0 |
| VCCATS | 3.3 | 0.007 | 0.158 | 0 | 0 |

## Table 10-4. PCH-H Measured $I_{CC}$ (Desktop and Server SkUs) (Sheet 2 of 2)

| Voltage Rail | Voltage (V) | S0 Iccmax Current[3] (A) | Sx Icc Idle Current[6] (mA) | Deep Sx Icc Idle Current (mA) | G3 (µA) |
|---|---|---|---|---|---|
| VCCHDA | 3.3 | 0.075 | 0.050 | 0 | 0 |
| | 1.8 | 0.075[8] | 0.050[8] | 0 | 0 |
| | 1.5 | 0.075[8] | 0.050[8] | 0 | 0 |
| VCCPRIM_3p3 | 3.3 | 0.171 | 0.543 | 0 | 0 |
| VCCDSW_3p3 | 3.3 | 0.204 | 3.41 | 3.41 | 0 |
| VCCRTCPRIM_3p3 | 3.3 | 0.350 mA | 0.227 | 0 | 0 |
| VCCRTC | 3.0 | 0.350 mA | 0.065 | 0.065 | 6<br>Notes 1, 2 |

***Notes:***
1. G3 state shown to provide an estimate of battery life.
2. Icc (RTC) data is taken with VCCRTC at 3.0V while the system is in a mechanical off (G3) state at room temperature.
3. Iccmax estimates assumes 110 °C.
4. The Iccmax value is a steady state current that can happen after respective power ok has asserted (or reset signal has de-asserted).
5. All data above are based on pre-silicon estimation and may be subject to change
6. Sx Icc Idle assumes PCH is idle and Intel ME is power gated.
7. N/A.
8. Sx Icc at 3.3V level is assumed as measured Sx Icc data at the 1.8 V and/or 1.5V level not measured.

## Table 10-5. PCH-H Measured $I_{CC}$ (H Mobile SkUs) (Sheet 1 of 2)

| Voltage Rail | Voltage (V) | S0 Iccmax Current[3] (A) | Sx Icc Idle Current[6] (mA) | Deep Sx Icc Idle Current (mA) | G3 (µA) |
|---|---|---|---|---|---|
| VCCPRIM_1p0 | 1.0 | 2.899 | 45.00 | 0 | 0 |
| VCCCLK1 | 1.0 | 0.021 | 0.150 | 0 | 0 |
| VCCCLK2 | 1.0 | 0.137 | 0.420 | 0 | 0 |
| VCCCLK3 | 1.0 | 0.050 | 0.170 | 0 | 0 |
| VCCCLK4 | 1.0 | 0.024 | 0.200 | 0 | 0 |
| VCCCLK5 | 1.0 | 0.010 | 0.690 | 0 | 0 |
| VCCMPHY_1p0 | 1.0 | See Table 10-6 | 4.00 | 0 | 0 |
| VCCHDAPLL_1p0 | 1.0 | 0.033 | 0.240 | 0 | 0 |
| VCCAMPHYPLL_1p0 | 1.0 | 0.080 | 0.400 | 0 | 0 |
| VCCAPLLEBB_1p0 | 1.0 | 0.075 | 0.150 | 0 | 0 |
| VCCMIPIPLL_1p0 | 1.0 | 0.030 | 0.200 | 0 | 0 |
| VCCUSB2PLL_1p0 | 1.0 | 0.012 | 0.492 | 0 | 0 |
| VCCPGPPA | 3.3 | 0.082 | 0.735 | 0 | 0 |
| | 1.8 | 0.082[8] | 0.735[8] | 0 | 0 |
| VCCPGPPBCH | 3.3 | 0.229 | 0.460 | 0 | 0 |
| | 1.8 | 0.229[8] | 0.460[8] | 0 | 0 |
| VCCPGPPD | 3.3 | 0.078 | 0.465 | 0 | 0 |
| | 1.8 | 0.078[8] | 0.465[8] | 0 | 0 |
| VCCPGPPEF | 3.3 | 0.114 | 0.500 | 0 | 0 |
| | 1.8 | 0.114[8] | 0.500[8] | 0 | 0 |

## Table 10-5. PCH-H Measured $I_{cc}$ (H Mobile SkUs) (Sheet 2 of 2)

| Voltage Rail | Voltage (V) | S0 Iccmax Current[3] (A) | Sx Icc Idle Current[6] (mA) | Deep Sx Icc Idle Current (mA) | G3 (µA) |
|---|---|---|---|---|---|
| VCCPGPPG | 3.3 | 0.065 | 0.400 | 0 | 0 |
| | 1.8 | 0.065[8] | 0.400[8] | 0 | 0 |
| VCCSPI | 3.3 | 0.029 | 0.217 | 0 | 0 |
| | 1.8 | 0.029[8] | 0.217[8] | 0 | 0 |
| VCCATS | 3.3 | 0.007 | 0.100 | 0 | 0 |
| VCCHDA | 3.3 | 0.060 | 0.300 | 0 | 0 |
| | 1.8 | 0.060[8] | 0.300[8] | 0 | 0 |
| | 1.5 | 0.060[8] | 0.300[8] | 0 | 0 |
| VCCPRIM_3p3 | 3.3 | 0.117 | 0.271 | 0 | 0 |
| VCCDSW_3p3 | 3.3 | 0.195 | 1.710 | 1.710 | 0 |
| VCCRTCPRIM_3p3 | 3.3 | 0.350 mA | 0.113 | 0 | 0 |
| VCCRTC | 3.0 | 0.350 mA | 0.036 | 0.036 | 6<br>Notes 1, 2 |

***Notes:***
1. G3 state shown to provide an estimate of battery life.
2. Icc (RTC) data is taken with VCCRTC at 3.0V while the system is in a mechanical off (G3) state at room temperature.
3. Iccmax estimates assumes 110 °C.
4. The Iccmax value is a steady state current that can happen after respective power ok has asserted (or reset signal has de-asserted).
5. All data above are based on pre-silicon estimation and may be subject to change
6. Sx Icc Idle assumes PCH is idle and Intel ME is power gated.
7. N/A.
8. Sx Icc at 3.3V level is assumed. Sx Icc data at the 1.8 V and/or 1.5V level not measured.

## Table 10-6. PCH-H VCCMPHY_1p0 Icc Adder Per HSIO Lane

| Icc (mA) | Details |
|---|---|
| 700 | All HSIO disabled. Assumes DMI x4 Running 100%. |
| 132 | Each USB 3.0 Port |
| 154 | Each PCIe Gen3 Lane |
| 54 | First SATA Gen3 Port |
| 132 | Each Additional SATA Gen3 Port |
| 102 | Each PCIe Gen2 Lane |
| 44 | GbE Port |

## Table 10-7. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 1 of 5)

| Type | Symbol | Parameter | Min | Max | Unit | Condition | Notes |
|---|---|---|---|---|---|---|---|
| Associated Signals[3]: HDA_BCLK, HDA_RST#, HDA_SDI0, HDA_SDI1, HDA_SDO, HDA_SYNC, DRAM_RESET#, GPD0 / BATLOW#, GPD1 / ACPRESENT, GPD10 / SLP_S5#, GPD11 / LANPHYPC, GPD2 / LAN_WAKE#, GPD3 / PWRBTN#, GPD4 / SLP_S3#, GPD5 / SLP_S4#, GPD6 / SLP_A#, GPD7 / RSVD, GPD8 / SUSCLK, GPD9 / SLP_WLAN#, GPP_A0 / RCIN# / ESPI_ALERT1#, GPP_A1 / LAD0 / ESPI_IO0, GPP_A10 / CLKOUT_LPC1, GPP_A11 /PME#, GPP_A12 / BMBUSY# / ISH_GP6 / SX_EXIT_HOLDOFF#, GPP_A13 / SUSWARN# / SUSPWRDNACK, GPP_A14 / SUS_STAT#/ ESPI_RESET#, GPP_A15 / SUSACK#, GPP_A16 / CLKOUT_48, GPP_A17 / ISH_GP7, GPP_A18 / ISH_GP0, GPP_A19 / ISH_GP1, GPP_A2 / LAD1 / ESPI_IO1, GPP_A20 / ISH_GP2, GPP_A21 / ISH_GP3, GPP_A22 / ISH_GP4, GPP_A23 / ISH_GP5, GPP_A3 / LAD2 / ESPI_IO2, GPP_A4 / LAD3 / ESPI_IO3, GPP_A5 / LFRAME# / ESPI_CS#, GPP_A6 / SERIRQ, GPP_A7 / PIRQA# / ESPI_ALERT0#, GPP_A8 / CLKRUN#, GPP_A9 / CLKOUT_LPC0 / ESPI_CLK, GPP_B0, GPP_B1, GPP_B11, GPP_B12 / SLP_S0#, GPP_B13 / PLTRST#, GPP_B14 / SPKR, GPP_B15 / GSPI0_CS#, GPP_B16 / GSPI0_CLK, GPP_B17 / GSPI0_MISO, GPP_B18 / GSPI0_MOSI, GPP_B19 / GSPI1_CS#, GPP_B2 / VRALERT#, GPP_B20 / GSPI1_CLK, GPP_B21 / GSPI1_MISO, GPP_B22 / GSPI1_MOSI, GPP_B3 / CPU_GP2, GPP_B4 / CPU_GP3, GPP_C10 / UART0_RTS#, GPP_C11 / UART0_CTS#, GPP_C12 / UART1_RXD / ISH_UART1_RXD, GPP_C13 / UART1_TXD / ISH_UART1_TXD, GPP_C14 / UART1_RTS# / ISH_UART1_RTS#, GPP_C15 / UART1_CTS# / ISH_UART1_CTS#, GPP_C20 / UART2_RXD, GPP_C21 / UART2_TXD, GPP_C22 / UART2_RTS#, GPP_C23 / UART2_CTS#, GPP_C8 / UART0_RXD, GPP_C9 / UART0_TXD, GPP_D0 , GPP_D1 , GPP_D10, GPP_D11, GPP_D12 / ISH_SPI_MOSI, GPP_D13 / ISH_UART0_RXD / SML0BDATA / I2C2_SDA, GPP_D14 / ISH_UART0_TXD / SML0BCLK / I2C2_SCL, GPP_D15 / ISH_UART0_RTS#, GPP_D16 / ISH_UART0_CTS#, GPP_D17 / DMIC_CLK1, GPP_D18 / DMIC_DATA1, GPP_D19 / DMIC_CLK0, GPP_D2 , GPP_D20 / DMIC_DATA0, GPP_D21, GPP_D22, GPP_D3, GPP_D5 / SSP0_SFRM, GPP_D6 / SSP0_TXD, GPP_D7 / SSP0_RXD, GPP_D8 / SSP0_SCLK, GPP_D9, GPP_F10 / SCLOCK, GPP_F11 / SLOAD, GPP_F12 / SDATAOUT1, GPP_F13 / SDATAOUT0, GPP_F14, GPP_F15 / USB2_OC4#, GPP_F16 / USB2_OC5#, GPP_F17 / USB2_OC6#, GPP_F18 / USB2_OC7#, GPP_F22, GPP_F23, GPP_F5 / DEVSLP3, GPP_F6 / DEVSLP4, GPP_F7 / DEVSLP5, GPP_F8 / DEVSLP6, GPP_F9 / DEVSLP7, GPP_G0 / GPP_G1 / GPP_G2 / GPP_G10 / GPP_G11 / GPP_G17 / ADR_COMPLETE, GPP_G18 / NMI#, GPP_G19 / SMI#, GPP_G20, GPP_G21, GPP_G22, GPP_G23, GPP_G3 / GPP_G4 / GPP_G5 / GPP_G6 / GPP_G7 / GPP_G8 / GPP_G9 / GPP_I0 / DDPB_HPD0, GPP_I1 / DDPC_HPD1, GPP_I2 / DDPD_HPD2, GPP_I3 / DDPE_HPD3, GPP_I4 / EDP_HPD, CL_RST#, SLP_LAN#, SLP_SUS#,SPI0_CLK, SPI0_CS0#, SPI0_CS1#, SPI0_IO2, SPI0_IO3, SPI0_MISO, SPI0_MOSI, SPI0_CS2#, SYS_PWROK, SYS_RESET#, WAKE#. | | | | | | | |
| **3.3V Operation** | | | | | | | |
| Input | $V_{IH}$ | Input High Voltage | 0.65 x $V_{CC}$ | $V_{CC}$ + 0.4 | V | | 1 |
| | $V_{IL}$ | Input Low Voltage | -0.5 | 0.35 x $V_{CC}$ | V | | 2 |
| | $I_{IL}$ | Input Leakage Current | -10 | 10 | µA | | |
| | $C_{IN}$ | Input Pin Capacitance | | 3 | pF | | |
| Output | $V_{OH}$ | Output High Voltage | 0.9 * $V_{CC}$ | $V_{CC}$ | V | $I_{oh}$=0.5mA | 4 |
| | $V_{OL}$ | Output Low Voltage | | 0.4 | V | $I_{ol}$=-4mA | 4 |
| | $R_{pu}$ | WPU Resistance | 5K-30%<br>20K-30% | 5K+30%<br>20K+30% | Ω | | |
| | $R_{pd}$ | WPD Resistance | 5K-30%<br>20K-30% | 5K+30%<br>20K+30% | Ω | | |
| **1.8V Operation** | | | | | | | |
| Input | $V_{IH}$ | Input High Voltage | 0.65 x $V_{CC}$ | $V_{CC}$ + 0.4 | V | | |
| | $V_{IL}$ | Input Low Voltage | -0.5 | 0.35 x $V_{CC}$ | V | | |
| | $I_{IL}$ | Input Leakage Current | -10 | 10 | µA | | |
| | $C_{IN}$ | Input Pin Capacitance | | 3 | pF | | |

## Table 10-7. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 2 of 5)

| Type | Symbol | Parameter | Min | Max | Unit | Condition | Notes |
|---|---|---|---|---|---|---|---|
| Output | $V_{OL}$ | Output Low Voltage | | 0.4 x $V_{CC}$ | V | $I_{ol}$=-4mA | 4 |
| | $R_{pu}$ | WPU Resistance | 5K-30%<br>20K-30% | 5K+30%<br>20K+30% | Ω | | |
| | $R_{pd}$ | WPD Resistance | 5K-30%<br>20K-30% | 5K+30%<br>20K+30% | Ω | | |
| ***Notes:***<br>1. VIH for LPC=0.5*VCC and VIH for HD Audio =0.6*VCC (*1.5V supply operation).<br>2. VIL for LPC=0.3*VCC and VIH for HD Audio =0.4*VCC (*1.5V supply operation).<br>3. For GPIO supported voltages, refer to GPIO chapter.<br>4. Each GPIO pin can support 3mA Ioh/Iol Max. | | | | | | | |
| | | | | | | | |
| Associated Signals[1]: GPP_B10 / SRCCLKREQ5#, GPP_B23 / SML1ALERT# / PCHHOT#, GPP_B5 / SRCCLKREQ0#, GPP_B6 / SRCCLKREQ1#, GPP_B7 / SRCCLKREQ2#, GPP_B8 / SRCCLKREQ3#, GPP_B9 / SRCCLKREQ4#, GPP_C0 / SMBCLK, GPP_C1 / SMBDATA, GPP_C16 / I2C0_SDA, GPP_C17 / I2C0_SCL, GPP_C18 / I2C1_SDA, GPP_C19 / I2C1_SCL, GPP_C2 / SMBALERT#, GPP_C3 / SML0CLK, GPP_C4 / SML0DATA, GPP_C5 / SML0ALERT#, GPP_C6 / SML1CLK, GPP_C7 / SML1DATA, GPP_D23 / ISH_I2C2_SCL / ISH_I2C3_SCL, GPP_D4 / ISH_I2C2_SDA / ISH_I2C3_SDA, GPP_F19 / eDP_VDDEN, GPP_F20 / eDP_BKLTEN, GPP_F21 / eDP_BKLTCTL, GPP_G12 / GSXDOUT, GPP_G13 / GSXSLOAD, GPP_G14 / GSXDIN, GPP_G15 / GSXSRESET#, GPP_G16 / GSXCLK, GPP_H0 / SRCCLKREQ6#, GPP_H1 / SRCCLKREQ7#, 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_H2 / SRCCLKREQ8#, GPP_H20 / ISH_I2C0_SCL, GPP_H21 / ISH_I2C1_SDA, GPP_H22 / ISH_I2C1_SCL, GPP_H23, GPP_H3 / SRCCLKREQ9#, GPP_H4 / SRCCLKREQ10#, GPP_H5 / SRCCLKREQ11#, GPP_H6 / SRCCLKREQ12#, GPP_H7 / SRCCLKREQ13#, GPP_H8 / SRCCLKREQ14#, GPP_H9 / SRCCLKREQ15#, GPP_I10 / DDPD_CTRLDATA, GPP_I5 / DDPB_CTRLCLK, GPP_I6 / DDPB_CTRLDATA, GPP_I7 / DDPC_CTRLCLK, GPP_I8 / DDPC_CTRLDATA, GPP_I9 / DDPD_CTRLCLK. | | | | | | | |
| **3.3V Operation** | | | | | | | |
| Input | $V_{IH}$ | Input High Voltage | 0.65 x $V_{CC}$ | $V_{CC}$ + 0.4 | V | | |
| | $V_{IL}$ | Input Low Voltage | -0.5 | 0.35 x $V_{CC}$ | V | | |
| | $I_{IL}$ | Input Leakage Current | -10 | 10 | µA | | |
| | $C_{IN}$ | Input Pin Capacitance | | 3.5 | pF | | |
| Output | $V_{OL}$ | Output Low Voltage | | 0.4 | V | $I_{ol}$=-4mA | 2 |
| | $R_{pu}$ | WPU Resistance | 5K-30%<br>20K-30% | 5K+30%<br>20K+30% | Ω | | |
| | $R_{pd}$ | WPD Resistance | 5K-30%<br>20K-30% | 5K+30%<br>20K+30% | Ω | | |
| **1.8V Operation** | | | | | | | |
| Input | $V_{IH}$ | Input High Voltage | 0.70 x $V_{CC}$ | $V_{CC}$ + 0.4 | V | | |
| | $V_{IL}$ | Input Low Voltage | -0.5 | 0.3 x $V_{CC}$ | V | | |
| | $I_{IL}$ | Input Leakage Current | -10 | 10 | µA | | |
| | $C_{IN}$ | Input Pin Capacitance | | 3.5 | pF | | |
| Output | $V_{OH}$ | Output High Voltage | 0.9 x $V_{CC}$ | | V | $I_{oh}$=0.5mA | 2 |
| | $V_{OL}$ | Output Low Voltage | | 0.4 | V | $I_{ol}$=-4mA | 2 |
| | $R_{pu}$ | WPU Resistance | 5K-30%<br>20K-30% | 5K+30%<br>20K+30% | Ω | $V_{pad}$=$V_{CC}$/2 | |
| | $R_{pd}$ | WPD Resistance | 5K-30%<br>20K-30% | 5K+30%<br>20K+30% | Ω | $V_{pad}$=$V_{CC}$/2 | |

## Table 10-7. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 3 of 5)

| Type | Symbol | Parameter | Min | Max | Unit | Condition | Notes |
|---|---|---|---|---|---|---|---|
| ***Notes:***<br>1. For GPIO supported voltages, refer to GPIO chapter.<br>2. Each GPIO pin can support 3mA Ioh/Iol Max. | | | | | | | |
| | | | | | | | |
| Associated Signals[1]: GPP_E0 / SATAXPCIE0 / SATAGP0, GPP_E1 / SATAXPCIE1 / SATAGP1, GPP_E2 / SATAXPCIE2 / SATAGP2, GPP_E3 / CPU_GP0, GPP_E4 / DEVSLP0, GPP_E5 / DEVSLP1, GPP_E6 / DEVSLP2, GPP_E7 / CPU_GP1, GPP_E8 / SATALED#, GPP_E10 / USB2_OC1#, GPP_E11 / USB2_OC2#, GPP_E12 / USB2_OC3#, GPP_E9 / USB2_OC0#, GPP_F0 / SATAXPCIE3 / SATAGP3, GPP_F1 / SATAXPCIE4 / SATAGP4, GPP_F2 / SATAXPCIE5 / SATAGP5, GPP_F3 / SATAXPCIE6 / SATAGP6, GPP_F4 / SATAXPCIE7 / SATAGP7 | | | | | | | |
| **3.3V Operation** | | | | | | | |
| Input | $V_{IH}$ | Input High Voltage | 0.65 x $V_{CC}$ | $V_{CC}$ + 0.4 | V | | |
| | $V_{IL}$ | Input Low Voltage | -0.5 | 0.35 x $V_{CC}$ | V | | |
| | $I_{IL}$ | Input Leakage Current | -10 | 10 | µA | | |
| | $C_{IN}$ | Input Pin Capacitance | | 3 | pF | | |
| Output | $V_{OH}$ | Output High Voltage | 0.9 x $V_{CC}$ | $V_{CC}$ | V | $I_{oh}$=0.5mA | 2 |
| | $V_{OL}$ | Output Low Voltage | | 0.1 x VCC | V | $I_{ol}$=-1.5mA | 2 |
| | $R_{pu}$ | WPU Resistance | 5K-30%<br>20K-30% | 5K+30%<br>20K+30% | Ω | | |
| | $R_{pd}$ | WPD Resistance | 5K-30%<br>20K-30% | 5K+30%<br>20K+30% | Ω | | |
| **1.8V Operation** | | | | | | | |
| Input | $V_{IH}$ | Input High Voltage | 0.65 x $V_{CC}$ | $V_{CC}$ | V | | |
| | $V_{IL}$ | Input Low Voltage | -0.5 | 0.35 x $V_{CC}$ | V | | |
| | $I_{IL}$ | Input Leakage Current | -10 | 10 | µA | | |
| | $C_{IN}$ | Input Pin Capacitance | | 3 | pF | | |
| Output | $V_{OL}$ | Output Low Voltage | | 0.4 | V | $I_{ol}$=-4mA | 2 |
| | $R_{pu}$ | WPU Resistance | 5K-30%<br>20K-30% | 5K+30%<br>20K+30% | Ω | $V_{pad}$=$V_{CC}$/2 | |
| | $R_{pd}$ | WPD Resistance | 5K-30%<br>20K-30% | 5K+30%<br>20K+30% | Ω | $V_{pad}$=$V_{CC}$/2 | |
| ***Notes:***<br>1. For GPIO supported voltages, refer to GPIO chapter.<br>2. Each GPIO pin can support 3mA Ioh/Iol Max. | | | | | | | |
| | | | | | | | |

## Table 10-7. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 4 of 5)

| Type | Symbol | Parameter | Min | Max | Unit | Condition | Notes |
|---|---|---|---|---|---|---|---|
| Associated Signals: DISPA_BCLK, DISPA_SDI, DISPA_SDO, PROCPWRGD, ITP_PMODE, JTAG_TCK, JTAG_TDI, JTAG_TDO, JTAG_TMS, JTAGX, PECI, PLTRST_CPU#, PM_DOWN, PM_SYNC, PRDY#, PREQ#, THERMTRIP#, PCH_TRIGIN, PCH_TRIGOUT. | | | | | | | |
| Input | $V_{IH}$ | Input High Voltage | PECI: 0.725 * $V_{CC}$ JTAG: 0.8 * $V_{CC}$ CMOS: 0.7 * $V_{CC}$ iDISPLAY: 0.65 * $V_{CC}$ | $V_{CC}$ + 0.25 | V | | |
| | $V_{IL}$ | Input Low Voltage | -0.5 | PECI: 0.275 * $V_{CC}$ JTAG: 0.51 * $V_{CC}$ CMOS/ iDISPLAY: 0.3 * $V_{CC}$ | | | |
| | $I_{IL}$ | Input Leakage Current | -10 | 10 | µA | | |
| | $C_{IN}$ | Input Pin Capacitance | | 2 | pF | | |
| Output | $V_{OH}$ | Output High Voltage | PECI: 0.75 x $V_{CC}$ iDISPLAY: 0.9 x VCC | $V_{CC}$ | V | $I_{oh}$=-6mA | |
| | $V_{OL}$ | Output Low Voltage | | PECI: 0.25 x VCC iDISPLAY: 0.9 x VCC | V | $I_{ol}$=.5mA | |
| | $R_{pu}$ | WPU Resistance | 1K-30% 20K-30% | 1K+30% 20K+30% | Ω | | |
| | $R_{pd}$ | WPD Resistance | 1K-30% 20K-30% | 1K+30% 20K+30% | Ω | | |
| | | | | | | | |
| **Associated Signals: CL_DATA, CL_CLK** | | | | | | | |
| | CL_$V_{Ref}$ | Supply Voltage Reference | 0.392 | 0.408 | V | | |
| Input | $V_{IH}$ | Input High Voltage | CL_$V_{ref}$ + 0.075 | | V | | |
| | $V_{IL}$ | Input Low Voltage | | CL_$V_{ref}$ - 0.075 | V | | |
| | $I_{IL}$ | Input Leakage Current | -10 | 10 | µA | | |
| | $C_{IN}$ | Input Pin Capacitance | | 2 | pF | | |
| Output | $V_{OH}$ | Output High Voltage | 0.61 | 0.98 | V | $R_{load}$ = 100 Ohm to GND | 1 |
| | $V_{OL}$ | Output Low Voltage | 0 | 0.15 | V | $I_{ol}$=1mA | |
| | $R_{pu}$ | WPU Resistance | 20K-30% | 20K+30% | Ω | | |
| | $R_{pd}$ | WPD Resistance | 20K-30% | 20K+30% | Ω | | |

## Table 10-7. Single-Ended Signal DC Characteristics as Inputs or Outputs (Sheet 5 of 5)

| Type | Symbol | Parameter | Min | Max | Unit | Condition | Notes |
|---|---|---|---|---|---|---|---|
| ***Notes:***<br>1. The $V_{OH}$ specification does not apply to open-collector or open-drain drivers. Signals of this type must have an external pull-up resistor, and that is what determines the high-output voltage level.<br>2. Input characteristics apply when a signal is configured as Input or to signals that are only Inputs. Output characteristics apply when a signal is configured as an Output or to signals that are only Outputs. | | | | | | | |

## Table 10-8. Single-Ended Signal DC Characteristics as Inputs or Outputs

| Type | Symbol | Parameter | Minimum | Maximum | Unit | Condition | Notes |
|---|---|---|---|---|---|---|---|
| **Associated Signals: INTRUDER#, RSMRST#, PCH_PWROK, DSW_PWROK, SRTCRST#** | | | | | | | |
| Input | $V_{IH}$ | Input High Voltage | 0.65 x $V_{CCRTC}$ | $V_{CCRTC}$+0.5 | V | | 4, 6 |
| | $V_{IL}$ | Input Low Voltage | -0.5 | 0.3 x $V_{CCRTC}$ | V | | 6 |
| | | | | | | | |
| **Associated Signals: RTCRST#** | | | | | | | |
| Input | $V_{IH}$ | Input High Voltage | 0.75 x $V_{CCRTC}$ | $V_{CCRTC}$+0.5 | V | | 4, 5, 6 |
| | $V_{IL}$ | Input Low Voltage | -0.5 | 0.4 x $V_{CCRTC}$ | V | | 6 |
| | | | | | | | |
| **Associated Signals: RTCX1#** | | | | | | | |
| Input | $V_{IH}$ | Input High Voltage | 0.8 | 1.2 | V | | |
| | $V_{IL}$ | Input Low Voltage | -0.5 | 0.1 | V | | |
| | | | | | | | |
| **Associated Signals: XTAL24_IN** | | | | | | | 3 |
| Input | $V_{IH}$ | Input High Voltage | 0.8 | 1.2 | V | | |
| | $V_{IL}$ | Input Low Voltage | -0.2 | 0.2 | V | | |
| ***Notes:***<br>1. The $V_{OH}$ specification does not apply to open-collector or open-drain drivers. Signals of this type must have an external Pull-up resistor, and that is what determines the high-output voltage level.<br>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.<br>3. Vpk-pk minimum for XTAL24 = 500 mV<br>4. $V_{CCRTC}$ is the voltage applied to the $V_{CCRTC}$ well of the PCH. When the system is in G3 state, it is generally supplied by the coin cell battery. In S5 or greater state, it is supplied by VCCSUS3_3<br>5. $V_{IH}$ min should not be used as the reference point for T200 timing. See T200 specification for the measurement point detail<br>6. These buffers have input hysteresis. $V_{IH}$ levels are for rising edge transitions and $V_{IL}$ levels are for falling edge transitions. | | | | | | | |

## Table 10-9. Differential Signals Characteristics (Sheet 1 of 4)

| Symbol | Parameter | Minimum | Maximum | Unit | Conditions | Notes |
|---|---|---|---|---|---|---|
| **Associated Signals: PCIe*** | | | | | | 9, 10 |
| **Gen 1** | | | | | | |
| VTX-DIFF P-P | Differential Peak to Peak Output Voltage | 0.8 | 1.2 | V | | 1 |
| VTX-DIFF P-P - Low | Low power differential Peak to Peak Output Voltage | 0.4 | 1.2 | V | | |
| VTX_CM-ACp | TX AC Common Mode Output Voltage (2.5 GT/s) | — | 20 | mV | | |

## Table 10-9. Differential Signals Characteristics (Sheet 2 of 4)

| Symbol | Parameter | Minimum | Maximum | Unit | Conditions | Notes |
|---|---|---|---|---|---|---|
| ZTX-DIFF-DC | DC Differential TX Impedance | 80 | 120 | Ohm | | |
| VRX-DIFF p-p | Differential Input Peak to Peak Voltage | 0.12 | 1.2 | V | | 1 |
| VRX_CM-ACp | AC peak Common Mode Input Voltage | — | 150 | mV | | |
| **Gen 2** | | | | | | |
| VTX-DIFF P-P | Differential Peak to Peak Output Voltage | 0.8 | 1.2 | V | | |
| VTX-DIFF P-P - Low | Low power differential Peak to Peak Output Voltage | 0.4 | 1.2 | V | | |
| VTX_CM-Acp-p | TX AC Common Mode Output Voltage (5GT/s) | — | 100 | mV | | |
| ZTX-DIFF-DC | DC Differential TX Impedance | 80 | 120 | Ohm | | |
| VRX-DIFF p-p | Differential Input Peak to Peak Voltage | 0.12 | 1.2 | V | | |
| VRX_CM-ACp | AC peak Common Mode Input Voltage | — | 150 | mV | | |
| **Gen 3** | | | | | | |
| VTX-DIFF P-P | Differential Peak to Peak Output Voltage | 0.8 | 1.3 | V | | |
| VTX-DIFF P-P - Low | Low power differential Peak to Peak Output Voltage | 0.4 | 1.2 | V | | |
| VTX_CM-Acp-p | TX AC Common Mode Output Voltage (5GT/s) | — | 100 | mV | | |
| ZTX-DIFF-DC | DC Differential TX Impedance | 80 | 120 | Ohm | | |
| VRX-DIFF p-p | Differential Input Peak to Peak Voltage | Refer to Stressed Voltage Eye Parameters Table in PCIe* GEN3 industry specifications. | | | | |
| VRX_CM-ACp | AC peak Common Mode Input Voltage | — | 150 | mV | | |
| | | | | | | |
| **Associated Signals: SATA** | | | | | | |
| VIMIN-Gen1i | Minimum Input Voltage - 1.5Gb/s internal SATA | 325 | — | mVdiff p-p | | 2 |
| VIMAX-Gen1i | Maximum Input Voltage - 1.5Gb/s internal SATA | — | 600 | mVdiff p-p | | 2 |
| VIMIN-Gen1m | Minimum Input Voltage - 1.5Gb/s eSATA | 240 | — | mVdiff p-p | | 2 |
| VIMAX-Gen1m | Maximum Input Voltage - 1.5Gb/s eSATA | — | 600 | mVdiff p-p | | 2 |
| VIMIN-Gen2i | Minimum Input Voltage - 3.0Gb/s internal SATA | 275 | — | mVdiff p-p | | 2 |
| VIMAX-Gen2i | Maximum Input Voltage - 3.0Gb/s internal SATA | — | 750 | mVdiff p-p | | 2 |
| VIMIN-Gen2m | Minimum Input Voltage - 3.0Gb/s eSATA | 240 | — | mVdiff p-p | | 2 |
| VIMAX-Gen2m | Maximum Input Voltage - 3.0Gb/s eSATA | — | 750 | mVdiff p-p | | 2 |
| VIMIN-Gen3i | Minimum Input Voltage - 6.0Gb/s internal SATA | 240 | — | mVdiff p-p | | 2 |
| VIMAX-Gen3i | Maximum Input Voltage - 6.0Gb/s internal SATA | — | 1000 | mVdiff p-p | | 2 |
| VOMIN-Gen1i,m | Minimum Output Voltage 1.5Gb/s internal and eSATA | 400 | — | mVdiff p-p | | 3 |

## Table 10-9. Differential Signals Characteristics (Sheet 3 of 4)

| Symbol | Parameter | Minimum | Maximum | Unit | Conditions | Notes |
|---|---|---|---|---|---|---|
| VOMAX-Gen1i,m | Maximum Output Voltage 1.5Gb/s internal and eSATA | — | 600 | mVdiff p-p | | 3 |
| VOMIN-Gen2i,m | Minimum Output Voltage 3.0Gb/s internal and eSATA | 400 | — | mVdiff p-p | | 3 |
| VOMAX-Gen2i,m | Maximum Output Voltage 3.0Gb/s internal and eSATA | — | 700 | mVdiff p-p | | 3 |
| VOMIN-Gen3i | Minimum Output Voltage 6.0Gb/s internal SATA | 200 | — | mVdiff p-p | | 3 |
| VOMAX-Gen3i | Maximum Output Voltage 6.0Gb/s internal SATA | — | 900 | mVdiff p-p | | 3 |
| | | | | | | |
| **Associated Signals: USB 2.0** | | | | | | |
| VDI | Differential Input Sensitivity | 0.2 | — | V | | 4, 6 |
| VCM | Differential Common Mode Range | 0.8 | 2.5 | V | | 5, 6 |
| VSE | Single-Ended Receiver Threshold | 0.8 | 2 | V | | 6 |
| VCRS | Output Signal Crossover Voltage | 1.3 | 2 | V | | 6 |
| VOL | Output Low Voltage | — | 0.4 | V | Iol=5 mA | 6 |
| VOH | Output High Voltage | 3.3V – 0.5 | — | V | Ioh=-2mA | 6 |
| VHSSQ | HS Squelch Detection Threshold | 100 | 150 | mV | | 7 |
| VHSDSC | HS Disconnect Detection Threshold | 525 | 625 | mV | | 7 |
| VHSCM | HS Data Signaling Common Mode Voltage Range | -50 | 500 | mV | | 7 |
| VHSOI | HS Idle Level | -10 | 10 | mV | | 7 |
| VHSOH | HS Data Signaling High | 360 | 440 | mV | | 7 |
| VHSOL | HS Data Signaling Low | -10 | 10 | mV | | 7 |
| VCHIRPJ | Chirp J Level | 700 | 1100 | mV | | 7 |
| VCHIRPK | Chirp K Level | -900 | -500 | mV | | 7 |
| ***Note:*** | VDI VCM, VSE, VCRS, VOL, VOH are USB 2.0 FS/LS electrical characteristic. | | | | | |
| | | | | | | |
| **Associated Signals: USB 3.0** | | | | | | |
| VTX-DIFF-PP | Differential Peak to Peak Output Voltage | 0.8 | 1.2 | V | | |
| VTX-DIFF P-P - Low | Low power differential Peak to Peak Output Voltage | 0.4 | 1.2 | V | | 8 |
| VTX_CM-Acp-p | TX AC Common Mode Output Voltage (5GT/s) | — | 100 | mV | | |
| ZTX-DIFF-DC | DC Differential TX Impedance | 72 | 120 | Ohm | | |
| VRX-DIFF p-p | Differential Input Peak to Peak Voltage | 0.1 | 1.2 | V | | |
| VRX_CM-ACp | AC peak Common Mode Input Voltage | — | 150 | mV | | |
| | | | | | | |
| **Associated Signals: RTCX1** | | | | | | |
| Input | $V_{IH}$ | Input High Voltage | 0.8 | 1.2 | V | | |
| | $V_{IL}$ | Input Low Voltage | -0.5 | 0.1 | V | | |

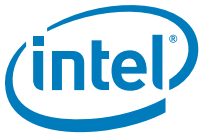
## Table 10-9. Differential Signals Characteristics (Sheet 4 of 4)

| Symbol | | Parameter | Minimum | Maximum | Unit | Conditions | Notes |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| **Associated Signals: _P/N, CLKOUT_CPUPCIBCLK_P/N, CLKOUT_CPUBCLK_P/N** | | | | | | | |
| Output | $V_{Swing}$ | Differential Output Swing | 300 | — | mV | | 12 |
| | $V_{Cross}$ | Crossing Point Voltage | 250 | 550 | mV | | 11,13,14 |
| | $V_{Cross_Delta}$ | Variation of VCROSS | — | 140 | mV | | 11,13,17 |
| | $V_{Max}$ | Max Output Voltage | — | 1.15 | V | | 11,15 |
| | $V_{Min}$ | Min Output Voltage | -0.3 | — | | | 11,16 |

***Notes:***
1. PCI Express mVdiff p-p = 2*|PCIE[x]_TXP – PCIE[x]_TXN|; PCI Express mVdiff p-p = 2*|CIE[x]_RXP – PCIE[x]_RXN|
2. SATA Vdiff, RX ($V_{IMAX}/V_{IMIN}$) is measured at the SATA connector on the receiver side (generally, the motherboard connector), where SATA mVdiff p-p = 2*|SATA[x]RXP – SATA[x]RXN|.
3. SATA Vdiff, tx ($V_{OMIN}/V_{OMAX}$) is measured at the SATA connector on the transmit side (generally, the motherboard connector), where SATA mVdiff p-p = 2*|SATA[x]TXP – SATA[x]TXN|
4. $V_{DI}$ = | USBPx[P] – USBPx[N] |
5. Includes VDI range
6. Applies to Low-Speed/Full-Speed USB
7. Applies to High-Speed USB 2.0.
8. USB 3.0 mVdiff p-p = 2*|USB3Rp[x] – USB3Rn[x]|; USB 3.0 mVdiff p-p = 2*|USB3Tp[x] – USB3Tn[x]|
9. For PCIe, GEN1, GEN and GEN3 correspond to the PCIe base specification revision 1, 2 and 3.
10. PCIE specifications are also applicable to the LAN port
11. Measurement taken from single-ended waveform on a component test board
12. Measurement taken from differential waveform on a component test board
13. $V_{Cross}$ is defined as the voltage where Clock = Clock#
14. Only applies to the differential rising edge (that is, Clock rising and Clock# falling)
15. The max. voltage including overshoot
16. The min. voltage including undershoot
17. The total variation of all $V_{Cross}$ measurements in any particular system. Note that this is a subset of $V_{Cross}$ MIN/MAX ($V_{Cross}$ absolute) allowed. The intent is to limit $V_{Cross}$ induced modulation by setting $V_{Cross_Delta}$ to be smaller than $V_{Cross}$ absolute.

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

| Symbol | Parameter | Min. | Nom. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|
| VCCPRIM_1p0 | Core Logic, SRAM, I/O Blocks, USB AFE, Processor Sideband, JTAG, Thermal Sensor Primary WellSP | 0.950 | 1.0 | 1.05 | V | 1 |
| VCCCLK1 | Clock Buffer 1 Primary Well | 0.950 | 1.0 | 1.05 | V | 1 |
| VCCCLK2 | Clock Buffer 2 Primary Well | 0.950 | 1.0 | 1.05 | V | 1 |
| VCCCLK3 | Clock Buffer 3 Primary Well | 0.950 | 1.0 | 1.05 | V | 1 |
| VCCCLK4 | Clock Buffer 4 Primary Well | 0.950 | 1.0 | 1.05 | V | 1 |
| VCCCLK5 | Clock Buffer 5 Primary Well | 0.950 | 1.0 | 1.05 | V | 1 |
| VCCAPLLEBB_1p0 | PCIe PLL EBB Primary Well | 0.950 | 1.0 | 1.05 | V | 1 |
| VCCAMPHYPLL_1p0 | Analog Supply for USB 3.0, PCIe Gen2, SATA and PCIe* Gen 3 PLL Primary Well | 0.950 | 1.0 | 1.05 | V | 1 |
| VCCAMPHY_1p0 | Mod-PHY Supply Primary Well | 0.950 | 1.0 | 1.05 | V | 1 |
| VCCHDAPLL_1p0 | Analog Supply for Audio PLL Primary Well | 0.950 | 1.0 | 1.05 | V | 1 |
| VCCMIPIPLL_1p0 | Analog Supply for MIPI PLL Primary Well | 0.950 | 1.0 | 1.05 | V | 1 |
| VCCUSB2PLL_1p0 | Analog Supply for USB 2.0 PLL Primary Well | 0.950 | 1.0 | 1.05 | V | 1 |
| VCCPGPPA | Group A Primary Well GPIOs | 3.13 | 3.3 | 3.46 | V | 1 |
| | | 1.71 | 1.8 | 1.89 | V | 1 |
| VCCPGPPBCH | Group B, C and H Primary Well GPIOs | 3.13 | 3.3 | 3.46 | V | 1 |
| | | 1.71 | 1.8 | 1.89 | V | 1 |



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

| Symbol | Parameter | Min. | Nom. | Max. | Unit | Notes |
|---|---|---|---|---|---|---|
| VCCPGPPD | Group D Primary Well GPIOs | 3.13 | 3.3 | 3.46 | V | 1 |
| | | 1.71 | 1.8 | 1.89 | V | 1 |
| VCCPGPPEF | Group E and F Primary Well GPIOs | 3.13 | 3.3 | 3.46 | V | 1 |
| | | 1.71 | 1.8 | 1.89 | V | 1 |
| VCCPGPPG | Group G Primary Well GPIOs | 3.13 | 3.3 | 3.46 | V | 1 |
| | | 1.71 | 1.8 | 1.89 | V | 1 |
| VCCSPI | SPI Primary Well | 3.13 | 3.3 | 3.46 | V | 1 |
| | | 1.71 | 1.8 | 1.89 | V | 1 |
| VCCATS | Thermal Sensor S0 Only Well | 3.13 | 3.3 | 3.46 | V | 1 |
| VCCHDA | Intel® HD Audio Supply Primary Well | 3.13 | 3.3 | 3.46 | V | 1 |
| | | 1.71 | 1.8 | 1.89 | V | 1 |
| | | 1.425 | 1.5 | 1.575 | V | 1 |
| VCCPRIM_3p3 | Primary Well for HVCMOS and display | 3.13 | 3.3 | 3.46 | V | 1 |
| VCCDSW_3p3 | Deep Sx Well for GPD and USB 2.0 | 3.13 | 3.3 | 3.46 | V | 1 |
| VCCRTCPRIM_3p3 | RTC Logic Primary Well | 3.13 | 3.3 | 3.46 | V | 1 |
| VCCRTC | RTC Well Supply | 2.0 | 3.0 | 3.2 | V | 1,2,3 |

***Notes:***
1. The I/O buffer supply voltage is measured at the PCH package pins. The tolerances shown in Table 10-10 are inclusive of all noise from DC up to 20 MHz. In testing, the voltage rails should be measured with a bandwidth limited oscilloscope that has a roll off of 3db/decade above 20 MHz.
2. Maximum Crystal ESR is 50 KOhms.
3. The initial VCCRTC voltage can exceed Vmax of 3.2 V (up to 3.47V) for ~1 week period without concerns about damage to the PCH.

## 10.5 AC Characteristics

### Table 10-11. PCI Express* Interface Timings (Sheet 1 of 2)

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

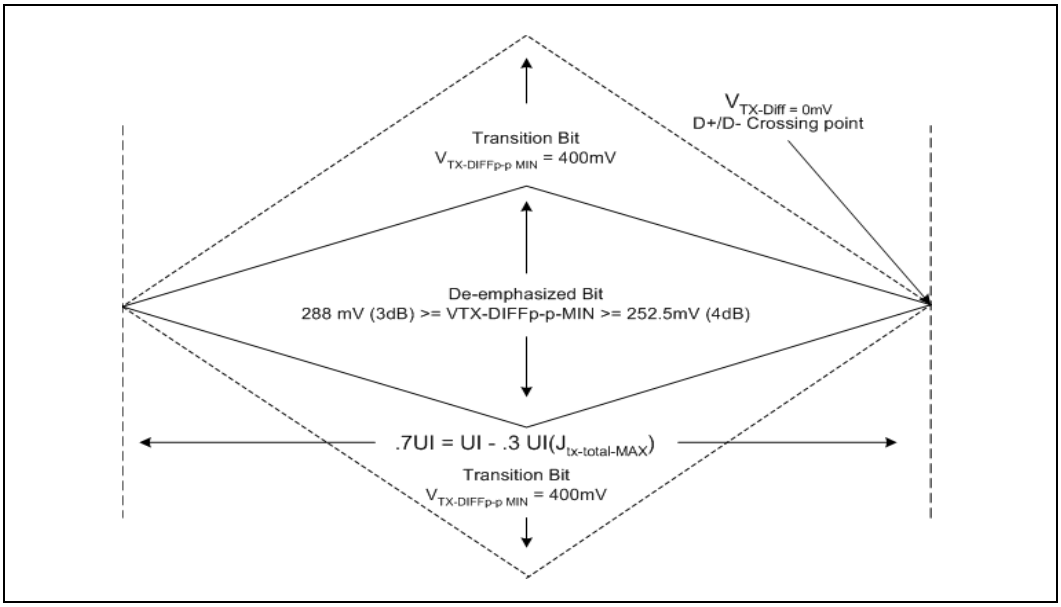
## Table 10-11. PCI Express* Interface Timings (Sheet 2 of 2)

| Symbol | Parameter | Min. | Max. | Unit | Figures | Notes |
|---|---|---|---|---|---|---|
| **Transmitter and Receiver Timings** | | | | | | |
| $T_{TX-EYE-MEDIAN-to-MAX-JITTER}$ (Gen 3) | Maximum time between the jitter median and maximum deviation from the median | 0.15 | — | UI | | |
| $T_{RX-EYE}$ (Gen 1) | Minimum Receiver Eye Width | 0.40 | — | UI | 10-2 | 3,4 |
| $T_{RX-EYE}$ (Gen 2) | Minimum Receiver Eye Width | 0.60 | — | UI | 10-2 | 3,4 |
| $T_{Min-Pulse}$ (Gen 2) | Instantaneous Pulse Width | 0.9 | — | UI | | |

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

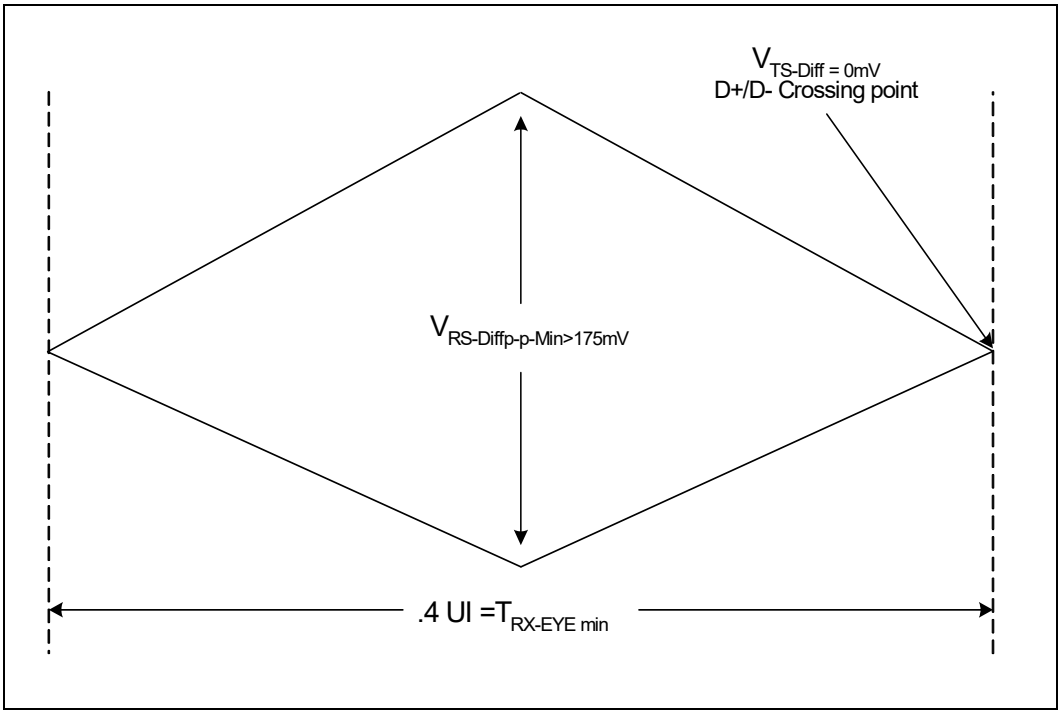
1. Specified at the measurement point into a timing and voltage compliance test load and measured over any 250 consecutive TX UIs. (also refer to the Transmitter compliance eye diagram)
2. A $T_{TX-EYE}$ = 0.70 UI provides for a total sum of deterministic and random jitter budget of $T_{TXJITTER-MAX}$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

## Figure 10-1. PCI Express* Transmitter Eye




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

## Figure 10-2. PCI Express* Receiver Eye




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

**Table 10-12. DDC Characteristics**

| Signal Group: eDP_VDDEN, eDP_BKLTEN, eDP_BKLTCTL, DDP[D:C:B]_CTRLCLK, DDP[D:C:B]_CTRLDATA | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Symbol** | **Parameter** | **Standard Mode** | **Fast Mode** | | **1 MHz** | | **Units** |
| | | **Max.** | **Min.** | **Max.** | **Min.** | **Max.** | |
| $F_{scl}$ | Operating Frequency | 100 | 0 | 400 | 0 | 1000 | KHz |
| $T_r$ | Rise Time[1] | 1000 | 20+0.1Cb[2] | 300 | — | 120 | ns |
| $T_f$ | Fall Time[1] | 300 | 20+0.1Cb[2] | 300 | — | 120 | ns |

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

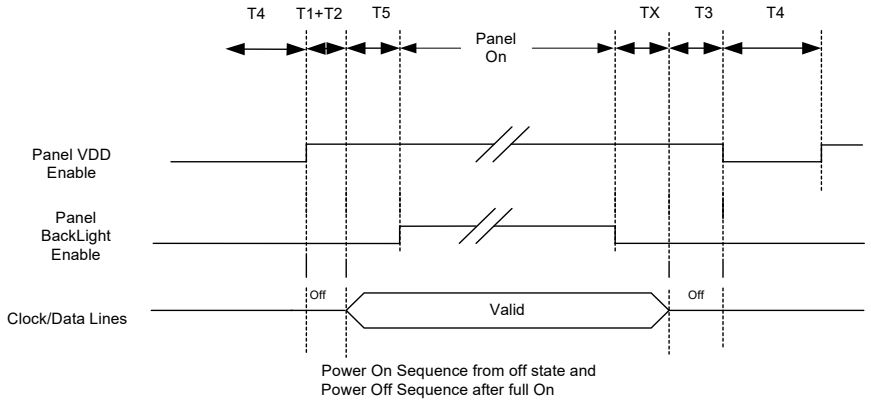
## 10.5.1 Panel Power Sequencing and Backlight Control

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

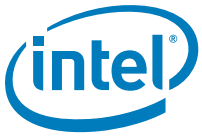
This section provides details for the power sequence timing relationship of the panel power, the backlight enable, and the eDP* data timing delivery. To meet the panel power timing specification requirements two signals, eDP_VDDEN and eDP_BKLTEN, are provided to control the timing sequencing function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

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

**Figure 10-3. Panel Power Sequencing**




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



## Table 10-13. DisplayPort* Hot-Plug Detect Interface

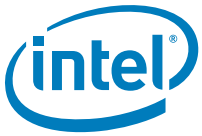
| Signal Group: DDPB_HPD0, DDPC_HPD1, DDPD_HPD2, DDPE_HPD3, eDP_HPD | | | | | | |
|---|---|---|---|---|---|---|
| **Symbol** | **Parameter** | **Min.** | **Max.** | **Unit** | **Figures** | **Notes** |
| Tir | Input Time Rise | 50 | 500 | ps | | |
| Tif | Input Time Fall | 50 | 500 | ps | | |
| Tidr | Input Delay Rise | 0.3 | 2.5 | ns | | |
| Tidf | Input Delay Fall | 0.3 | 2.5 | ns | | |

## Table 10-14. Clock Timings (Sheet 1 of 3)

| Sym | Parameter | Min. | Max. | Unit | Notes | Figure |
|---|---|---|---|---|---|---|
| **LPC Clock (CLKOUT_LPC[1:0])** | | | | | | |
| t1 | Period | 41.16 | 42.18 | ns | | 10-4 |
| t2 | High Time | 16.67 | 25.00 | ns | | 10-4 |
| t3 | Low Time | 16.67 | 25.00 | ns | | 10-4 |
| | Duty Cycle | 40 | 60 | % | | |
| | Jitter | — | 500 | ps | 8,9 | |
| | Flight Time (PCH to Device) | | 3 | ns | | |
| **CLKOUT_PCIE_P/N[15:0], CLKOUT_ITPXDP_[P,N]** | | | | | | |
| Period | Period SSC On | 9.849 | 10.201 | ns | | 10-5 |
| Period | Period SSC Off | 9.849 | 10.151 | ns | | 10-5 |
| DtyCyc | Duty Cycle | 40 | 60 | % | | 10-5 |
| V_Swing | Differential Output Swing | 300 | — | mV | | 10-5 |
| Slew_rise | Rising Edge Rate | 1.5 | 4 | V/ns | | 10-5 |
| Slew_fall | Falling Edge Rate | 1.5 | 4 | V/ns | | 10-5 |
| | Jitter | — | 150 | ps | 8,9,10 | |
| SSC | Spread Spectrum | 0 | 0.5 | % | 11 | |
| **CLKOUT_CPUNSSC_P/N** | | | | | | |
| Slew_rise | Rising Edge Rate | 1.3 | 4 | V/ns | | 10-5 |
| Slew_fall | Falling Edge Rate | 1.3 | 4 | V/ns | | 10-5 |
| Slew_var | Slew Rate Matching | — | 20 | % | | 10-5 |
| DtyCyc | Duty Cycle | 45 | 55 | % | | 10-5 |
| | Jitter | — | 6 | pS RMS | | 10-5 |
| **CLKOUT_CPUPCIBCLK_P/N, CLKOUT_CPUBCLK_P/N** | | | | | | |
| Slew_rise | Rising Edge Rate | 1.3 | 4 | V/ns | | 10-5 |
| Slew_fall | Falling Edge Rate | 1.3 | 4 | V/ns | | 10-5 |
| Slew_var | Slew Rate Matching | — | 20 | % | | 10-5 |
| DtyCyc | Duty Cycle | 40 | 60 | % | | 10-5 |
| | Jitter | — | 150 | ps | | 10-5 |

## Table 10-14. Clock Timings (Sheet 2 of 3)

| Sym | Parameter | Min. | Max. | Unit | Notes | Figure |
|---|---|---|---|---|---|---|
| **SMBus/SMLink Clock (SMBCLK, SML[1:0]CLK)** | | | | | | |
| $f_{smb}$ | Operating Frequency | 10 | 100 | KHz | | |
| t18 | High Time | 4.0 | 50 | µs | 2 | 10-6 |
| t19 | Low Time | 4.7 | — | µs | | 10-6 |
| t20 | Rise Time | — | 1000 | ns | | 10-6 |
| t21 | Fall Time | — | 300 | ns | | 10-6 |
| **SMLink[1,0] (SML[1:0]CLK) (Fast Mode: See note 15)** | | | | | | |
| $f_{smb}$ | Operating Frequency | 0 | 400 | KHz | | |
| t18_SMLFM | High Time | 0.6 | 50 | µs | 2 | 10-6 |
| t19_SMLFM | Low Time | 1.3 | — | µs | | 10-6 |
| t20_SMLFM | Rise Time | — | 300 | ns | | 10-6 |
| t21_SMLFM | Fall Time | — | 300 | ns | | 10-6 |
| **SMLink[1,0] (SML[1,0]CLK) (Fast Mode Plus: See note 17)** | | | | | | |
| $f_{smb}$ | Operating Frequency | 0 | 1000 | KHz | | |
| t18_SMLFMP | High Time | 0.26 | — | µs | 2 | 10-6 |
| t19_SMLFMP | Low Time | 0.5 | — | µs | | 10-6 |
| t20_SMLFMP | Rise Time | — | 120 | ns | | 10-6 |
| t21_SMLFMP | Fall Time | — | 120 | ns | | 10-6 |
| **HDA_BLK (Intel® High Definition Audio)** | | | | | | |
| $f_{HDA}$ | Operating Frequency | 24.0 | | MHz | | |
| | Frequency Tolerance | — | 100 | ppm | | |
| t26a | Input Jitter (refer to Clock Chip Specification) | — | 300 | ppm | | |
| t27a | High Time (Measured at 0.75 Vcc) | 18.75 | 22.91 | ns | | 10-4 |
| t28a | Low Time (Measured at 0.35 Vcc) | 18.75 | 22.91 | ns | | 10-4 |
| **Suspend Clock (SUSCLK)** | | | | | | |
| $f_{susclk}$ | Operating Frequency | 32 | | KHz | 4 | |
| t39 | High Time | 9.5 | — | µs | 4 | |
| t39a | Low Time | 9.5 | — | µs | 4 | |
| **XTAL24_IN/XTAL24_OUT** | | | | | | |
| ppm[12] | Crystal Tolerance cut accuracy maximum | 35 ppm(@ 25 °C ±3 °C) | | | | |
| ppm[12] | Temp Stability Maximum | 30 ppm(10 – 70 °C) | | | | |
| ppm[12] | Aging Maximum | 5 ppm | | | | |
| **CLKOUT_48 (Server/Workstation SKUs only)** | | | | | | |
| $f_{CLKOUT_48}$ | Operating Frequency | 48 | | MZ | | |
| | Period | 20.32 | 21.34 | nS | | 10-4 |
| | High Time | 7.02 | 12.51 | nS | | 10-4 |
| | Low Time | 6.63 | 12.30 | nS | | 10-4 |



## Table 10-14. Clock Timings (Sheet 3 of 3)

| Sym | Parameter | Min. | Max. | Unit | Notes | Figure |
|---|---|---|---|---|---|---|
| | Duty Cycle | 40 | 60 | % | | |
| | Rising Edge Rate | 1.0 | 4 | V/nS | | 10-4 |
| | Falling Edge Rate | 1.0 | 4 | V/nS | | 10-4 |
| | Jitter | — | 500 | pS | | |

***Notes:***
1. N/A
2. The maximum high time (t18 Max.) provides a simple ensured method for devices to detect bus idle conditions.
3. BCLK Rise and Fall times are measured from 10% VDD and 90% VDD.
4. SUSCLK duty cycle can range from 30% minimum to 70% maximum.
5. Edge rates in a system as measured from 0.8 – 2.0V.
6. The active frequency can be 5 MHz, 50 MHz, or 62.5 MHz depending on the interface speed. Dynamic changes of the normal operating frequency are not allowed.
7. Testing condition: 1 KΩ Pull-up to Vcc, 1 KΩ Pull-down and 10 pF Pull-down and 1/2 inch trace.
8. Jitter is specified as cycle-to-cycle as measured between two rising edges of the clock being characterized. Period minimum and maximum include cycle-to-cycle jitter and is also measured between two rising edges of the clock being characterized.
9. On all jitter measurements care should be taken to set the zero crossing voltage (for rising edge) of the clock to be the point where the edge rate is the fastest. Using a Math function = Average(Derivative(Ch1)) and set the averages to 64, place the cursors where the slope is the highest on the rising edge—usually this lower half of the rising edge. The reason this is defined is for users trying to measure in a system it is impossible to get the probe exactly at the end of the Transmission line with large Flip-Chip components. This results in a reflection induced ledge in the middle of the rising edge and will significantly increase measured jitter.
10. Phase jitter requirement: The designated outputs will meet the reference clock jitter requirements from the *PCI Express Base Specification*. The test is to be performed on a component test board under quiet conditions with all clock outputs on. Jitter analysis is performed using a standardized tool provided by the PCI SIG. Measurement methodology is defined in the Intel document "*PCI Express Reference Clock Jitter Measurements*". This is not for ITPXDP_P/N.
11. Spread Spectrum (SSC) is referenced to rising edge of the clock.
12. Total of crystal cut accuracy, frequency variations due to temperature, parasitics, load capacitance variations and aging is recommended to be less than 90 ppm.
13. Spread Spectrum (SSC) is referenced to rising edge of the clock.
14. Spread Spectrum (SSC) of 0.25% on CLKOUT_PCIE[7:0] and CLKOUT_PEG_[B:A] is used for WiMAX friendly clocking purposes.

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

## Figure 10-4. Clock Timing

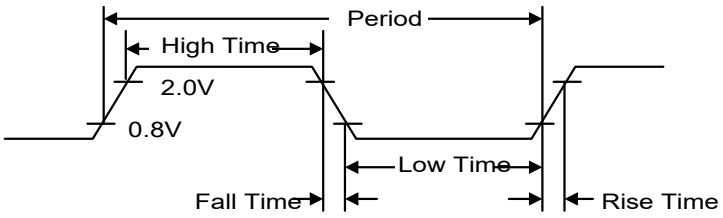
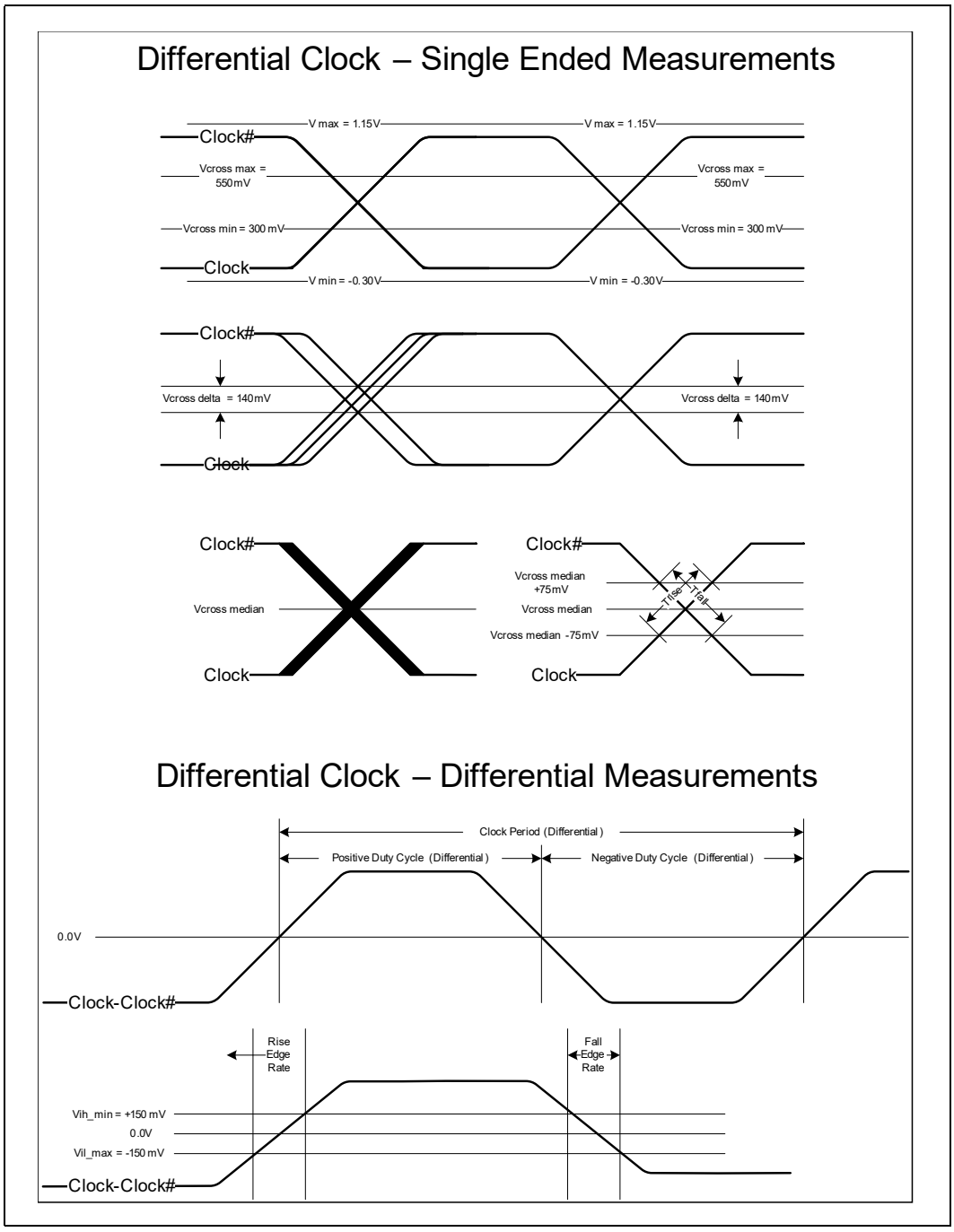
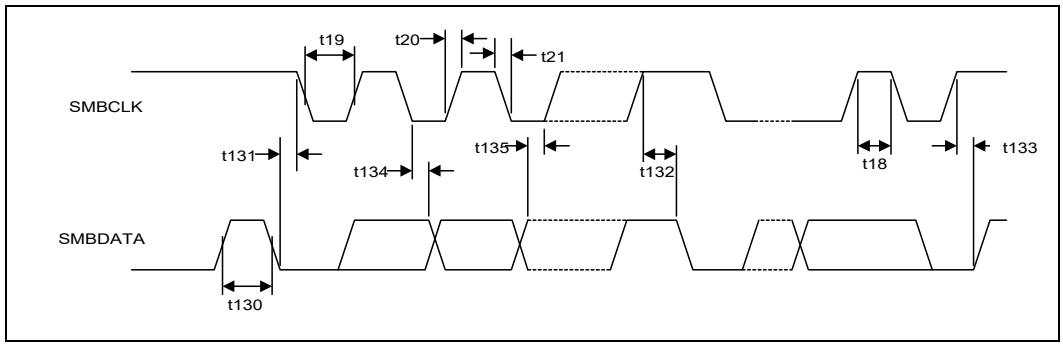


**Figure 10-5. Measurement Points for Differential Waveforms**

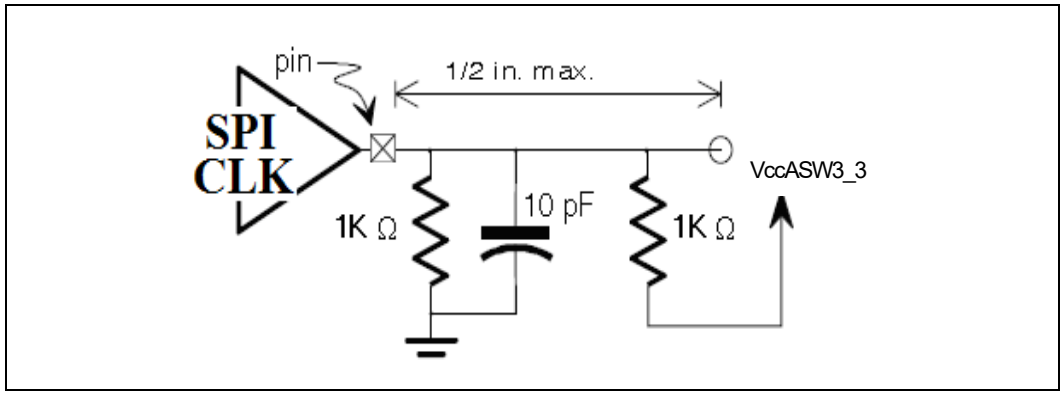


### Figure 10-6. SMBus/SMLink Transaction




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

### Figure 10-7. PCH Test Load




### Table 10-15. USB 2.0 Timing (Sheet 1 of 2)

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

## Table 10-15. USB 2.0 Timing (Sheet 2 of 2)

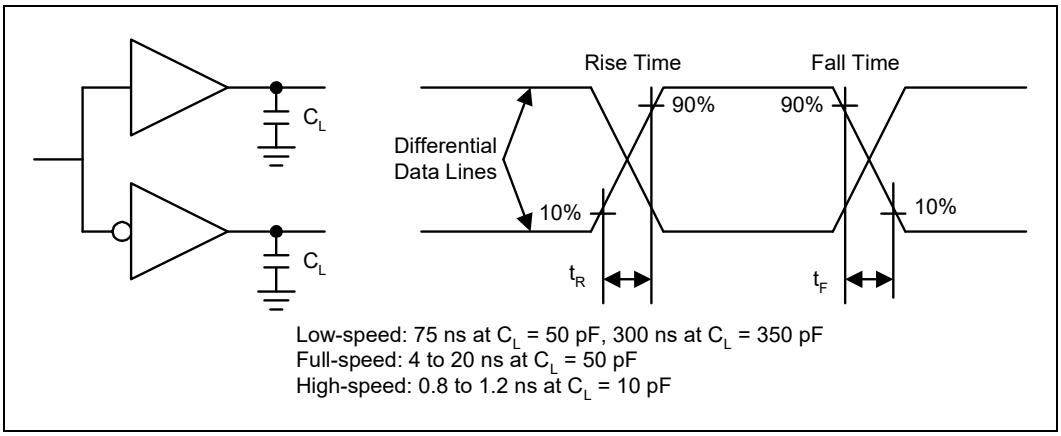
| Sym | Parameter | Min. | Max. | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| **Low-Speed Source (Note 8)** | | | | | | |
| t108 | USBPx+, USBPx – Driver Rise Time | 75 | 300 | ns | 1,6<br>$C_L$ = 200 pF<br>$C_L$ = 600 pF | 10-8 |
| t109 | USBPx+, USBPx – Driver Fall Time | 75 | 300 | ns | 1,6<br>$C_L$ = 200 pF<br>$C_L$ = 600 pF | 10-8 |
| t110 | Source Differential Driver Jitter<br>To Next Transition<br>For Paired Transitions | –25<br>–14 | 25<br>14 | ns<br>ns | 2,3 | 10-9 |
| t111 | Source SE0 interval of EOP | 1.25 | 1.50 | µs | 4 | 10-10 |
| t112 | Source Jitter for Differential Transition to SE0 Transition | –40 | 100 | ns | 5 | |
| t113 | Receiver Data Jitter Tolerance<br>- To Next Transition<br>- For Paired Transitions | –152<br>–200 | 152<br>200 | ns<br>ns | 3 | 10-9 |
| t114 | EOP Width: Receiver must accept EOP | 670 | — | ns | 4 | 10-10 |
| t115 | Width of SE0 interval during differential transition | — | 210 | ns | | |

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

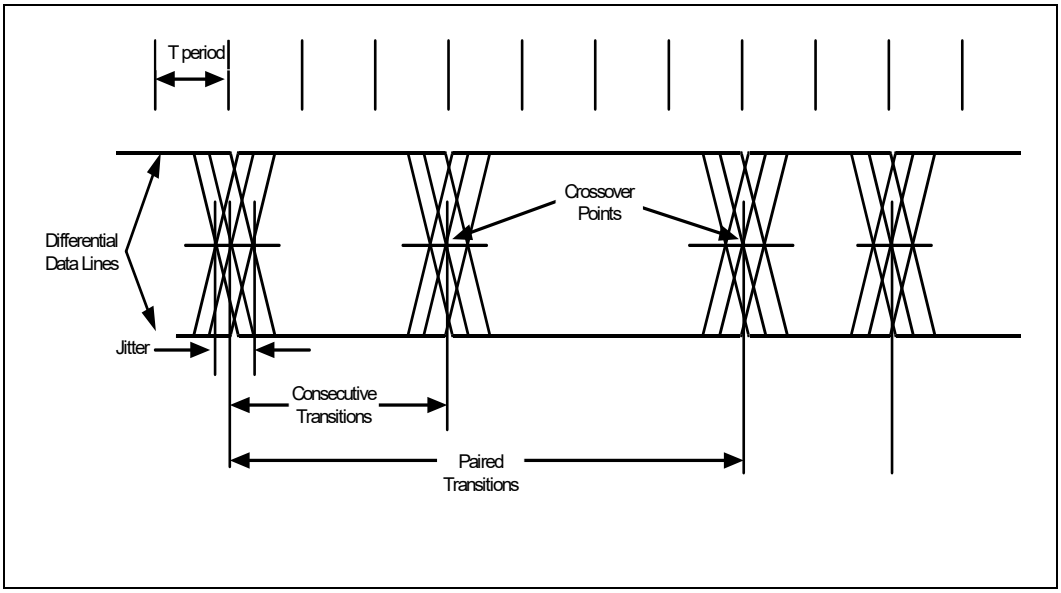
## Table 10-16. USB 3.0 Interface Transmit and Receiver Timings

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

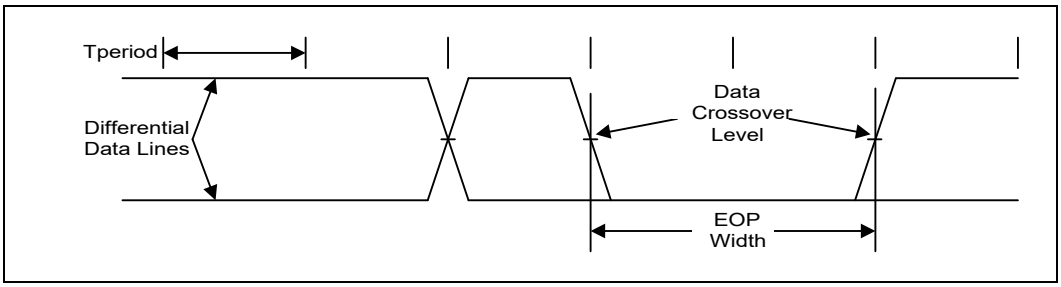
### Figure 10-8. USB Rise and Fall Times




### Figure 10-9. USB Jitter




### Figure 10-10.USB EOP Width




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

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

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

| Symbol | Parameter | Minimum | Maximum | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| t120gen3 | Rise Time | 0.2 | 0.48 | UI | 1 | |
| t121gen3 | Fall Time | 0.2 | 0.48 | UI | 2 | |
| t122 | TX differential skew | — | 20 | ps | | |
| t123 | COMRESET | 304 | 336 | ns | 3 | |
| t124 | COMWAKE transmit spacing | 101.3 | 112 | ns | 3 | |
| t125 | OOB Operating Data period | 646.67 | 686.67 | ns | 4 | |

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

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

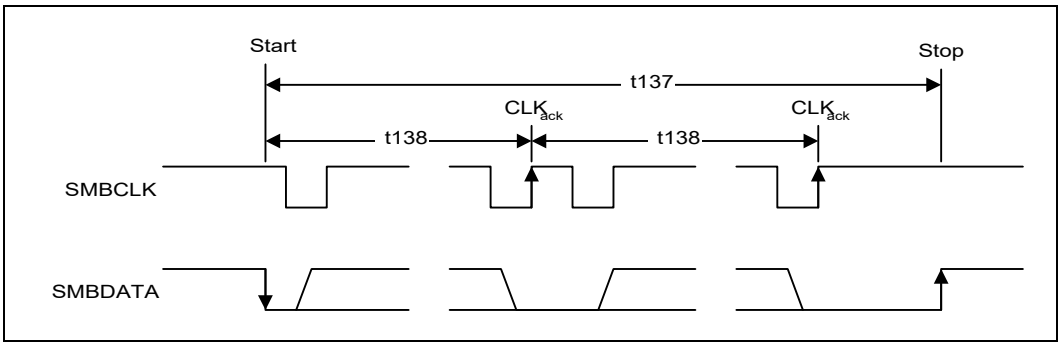
| Sym | Parameter | Min. | Max. | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| t130 | Bus Free Time Between Stop and Start Condition | 4.7 | — | µs | | 10-6 |
| t130SMLFM | Bus Free Time Between Stop and Start Condition | 1.3 | — | µs | 5 | 10-6 |
| t130SMLFMP | Bus Free Time Between Stop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

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

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

***Notes:***
1. A device will timeout when any clock low exceeds this value.
2. t137 is the cumulative time a slave device is allowed to extend the clock cycles in one message from the initial start to stop. If a slave device exceeds this time, it is expected to release both its clock and data lines and reset itself.
3. t138 is the cumulative time a master device is allowed to extend its clock cycles within each byte of a message as defined from start-to-ack, ack-to-ack, or ack-to-stop.
4. t134 has a minimum timing for $I^2C$ of 0 ns, while the minimum timing for SMBus/SMLINK is 300 ns.
5. Timings with the SMLFM designator apply only to SMLink[1,0] when operating in Fast Mode.

## Figure 10-11. SMBus/SMLink Timeout

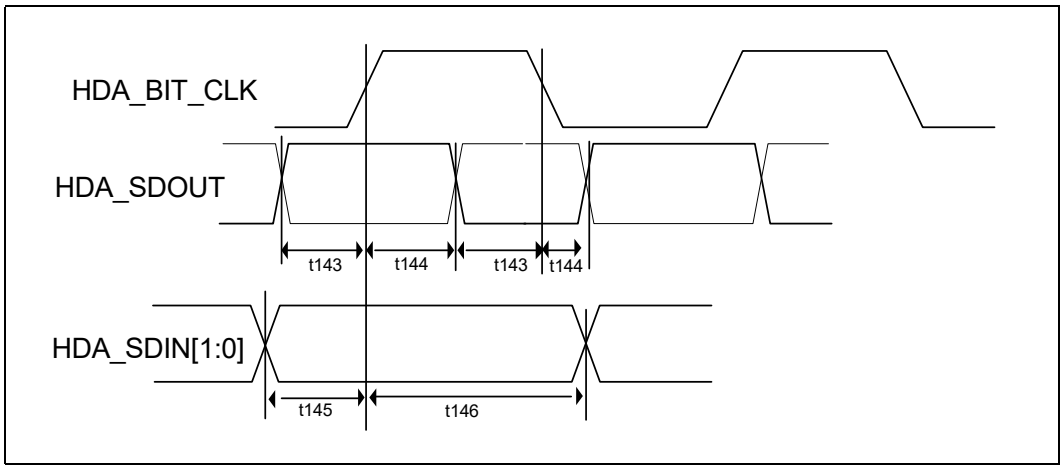



***Note:*** SMBCLK also refers to SML[1:0]CLK and SMBDATA also refers to SML[1:0]DATA in Figure 10-6.

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

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

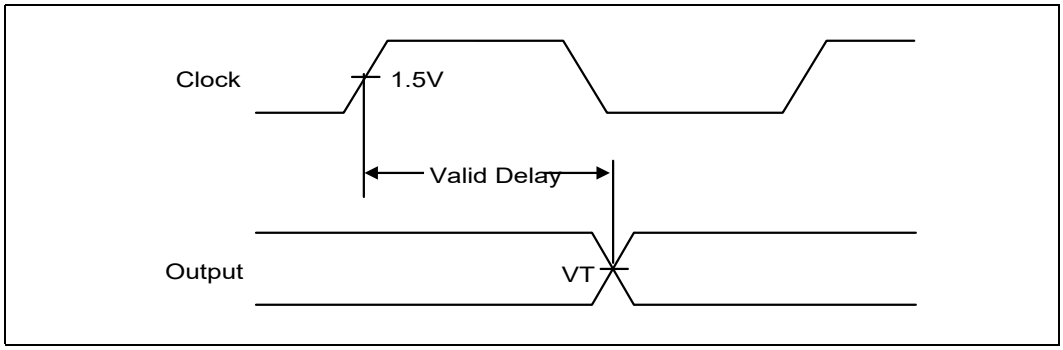
### Figure 10-12.Intel® High Definition Audio (Intel® HD Audio) Input and Output Timings



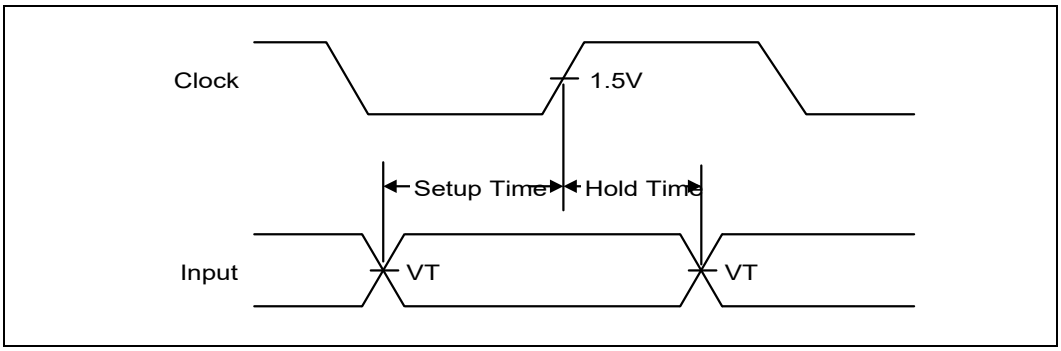

### Table 10-20. LPC Timing

| Sym | Parameter | Min. | Max. | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| t150 | LAD[3:0] Valid Delay from CLKOUT_LPC[1:0] Rising | 3 | 24.67 | ns | | 10-13 |
| t151 | LAD[3:0] Output Enable Delay from CLKOUT_LPC[1:0] Rising | 2 | — | ns | | 10-16 |
| t152 | LAD[3:0] Float Delay from CLKOUT_LPC[1:0] Rising | — | 28 | ns | | 10-15 |
| t153 | LAD[3:0] Setup Time to CLKOUT_LPC[1:0] Rising | 17.67 | — | ns | | 10-14 |
| t154 | LAD[3:0] Hold Time from CLKOUT_LPC[1:0] Rising | 2 | — | ns | | 10-14 |
| t157 | LFRAME# Valid Delay from CLKOUT_LPC[1:0] Rising | 3 | 24.67 | ns | | 10-13 |

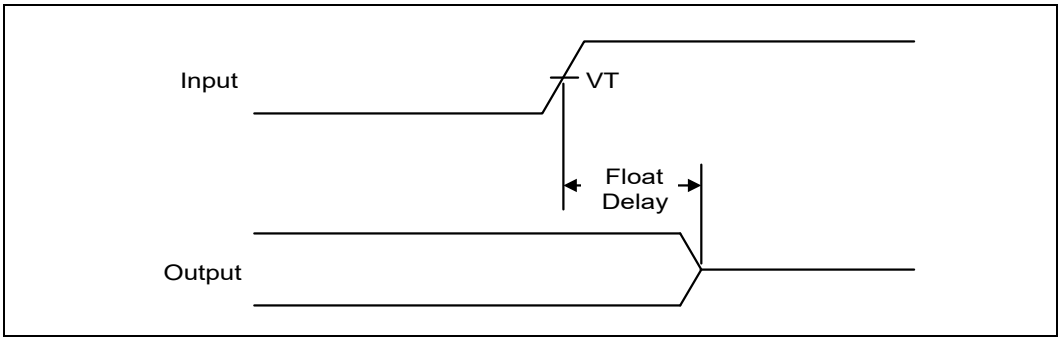
### Figure 10-13.Valid Delay from Rising Clock Edge



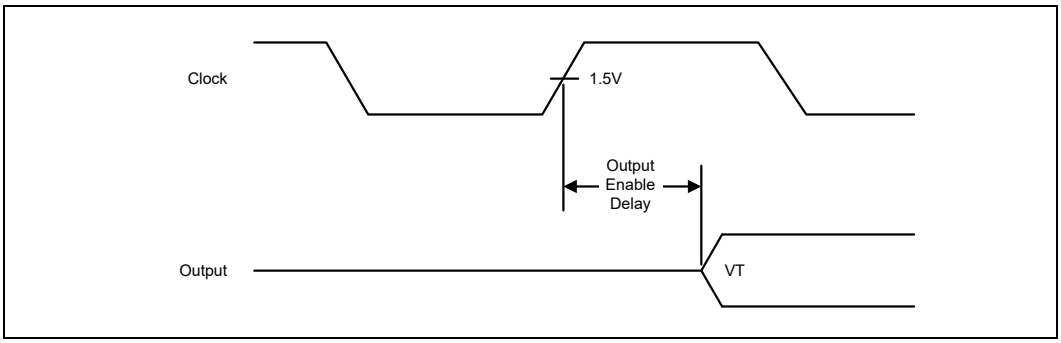
### Figure 10-14.Setup and Hold Times




### Figure 10-15.Float Delay



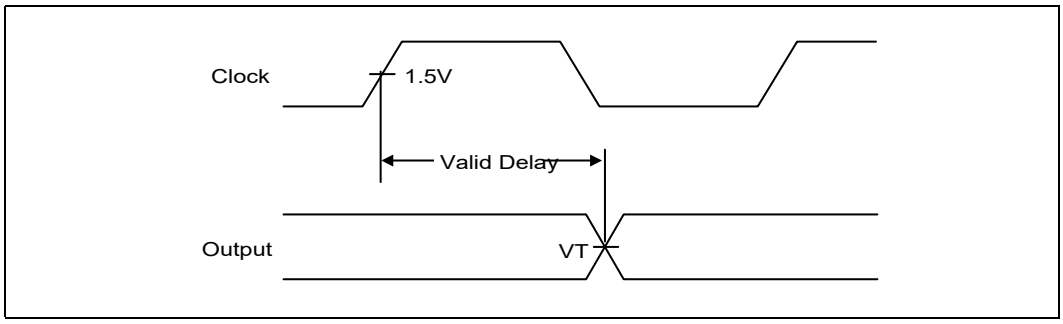

### Figure 10-16.Output Enable Delay



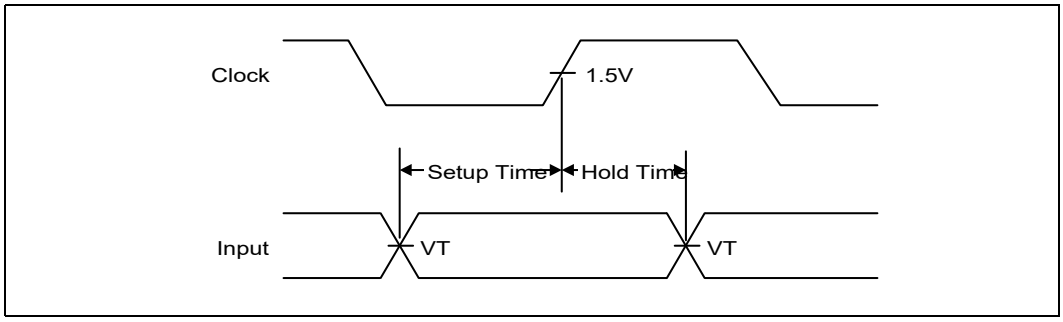

### Table 10-21. Miscellaneous Timings

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

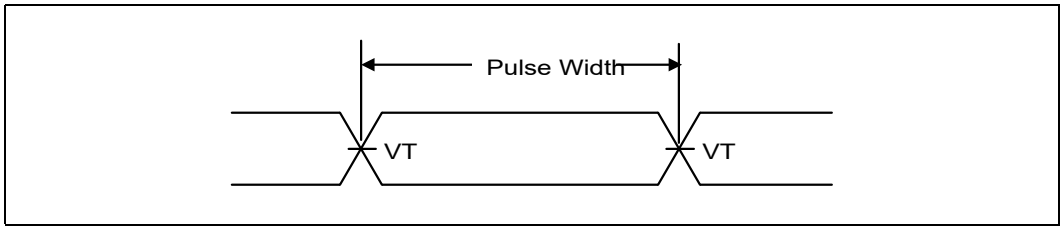
### Figure 10-17.Valid Delay from Rising Clock Edge




### Figure 10-18.Setup and Hold Times



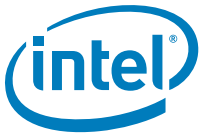

### Figure 10-19.Pulse Width




### Table 10-22. SPI Timings (17MHz)

| Symbol | Parameter | Min. | Max. | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| t180a | Serial Clock Frequency | 16.8 | 17.48 | MHz | 1 | |
| t183a | Tco of SPI MOSI and SPI I/O with respect to serial clock falling edge at the host | -5 | 13 | ns | | 10-20 |
| t184a | Setup of SPI MISO and SPI I/O with respect to serial clock falling edge at the host | 16 | — | ns | | 10-20 |
| t185a | Hold of SPI MISO and SPI I/O with respect to serial clock falling edge at the host | 0 | — | ns | | 10-20 |
| t186a | Setup of SPI CS# assertion with respect to serial clock rising edge at the host | 30 | — | ns | | 10-20 |
| t187a | Hold of SPI CS# assertion with respect to serial clock falling edge at the host | 30 | — | ns | | 10-20 |
| t188a | SPI CLK High time | 26.37 | — | ns | | 10-20 |
| t189a | SPI CLK Low time | 26.82 | — | ns | | 10-20 |

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



## Table 10-23. SPI Timings (30 MHz)

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

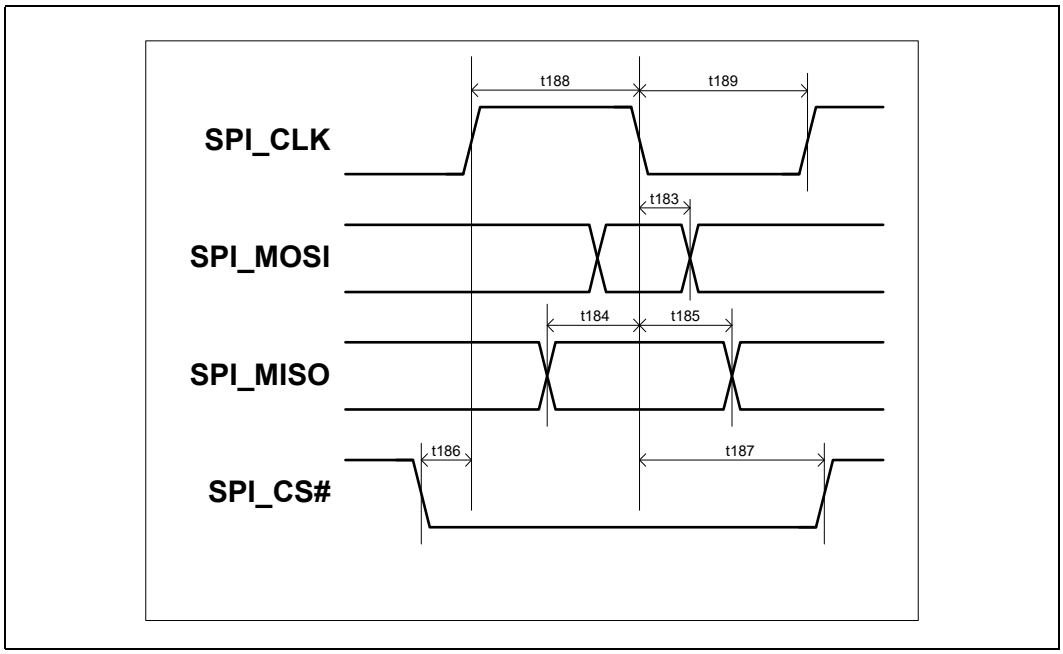
***Note:***
1. The typical clock frequency driven by the PCH is 30 MHz.
2. Measurement point for low time and high time is taken at 0.5(VccSPI).

## Table 10-24. SPI Timings (48 MHz)

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

***Note:***
1. Typical clock frequency driven by the PCH is 48 MHz.
2. When using 48 MHz mode ensure target flash component can meet t188c and t189c specifications. Measurement should be taken at a point as close as possible to the package pin.
3. Measurement point for low time and high time is taken at 0.5(VccSPI).

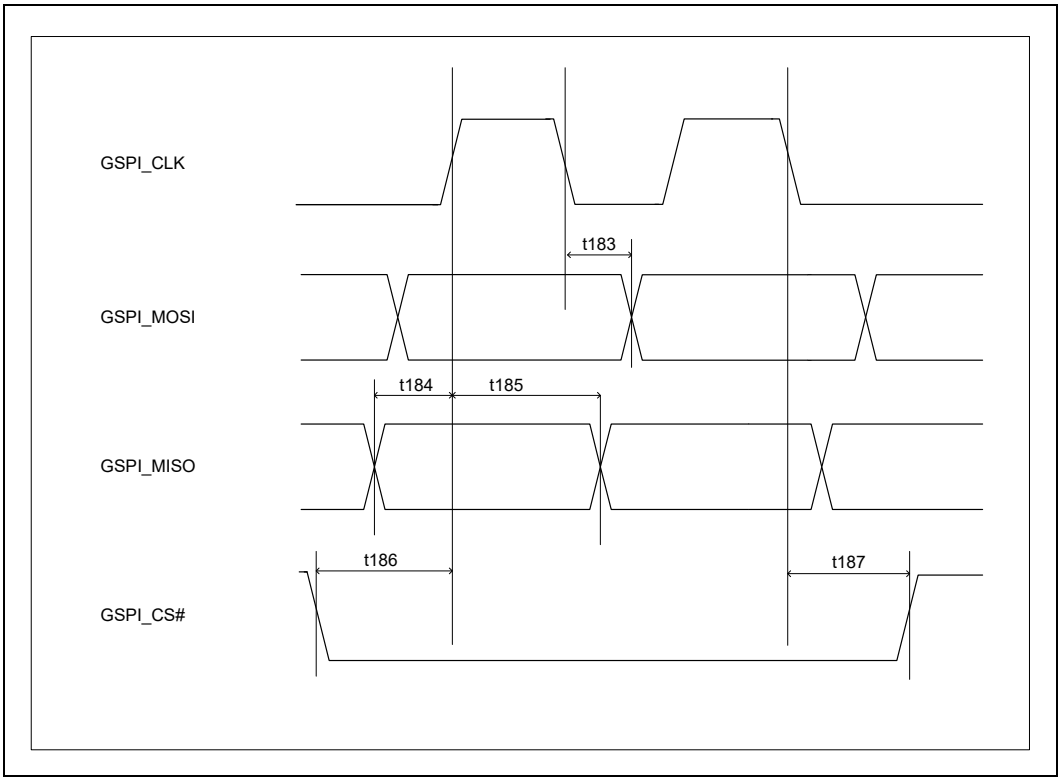
## Figure 10-20.SPI Timings




## Table 10-25. GSPI Timings (20 MHz)

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

## Figure 10-21. GSPI Timings

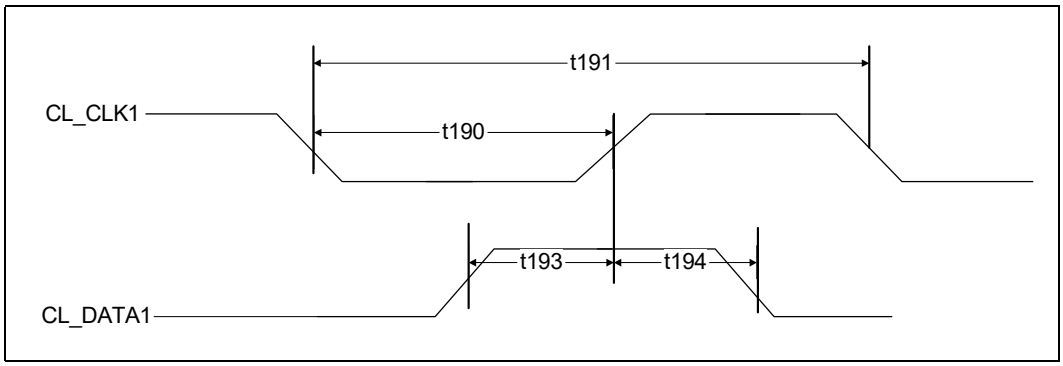



## Table 10-26. Controller Link Receive Timings

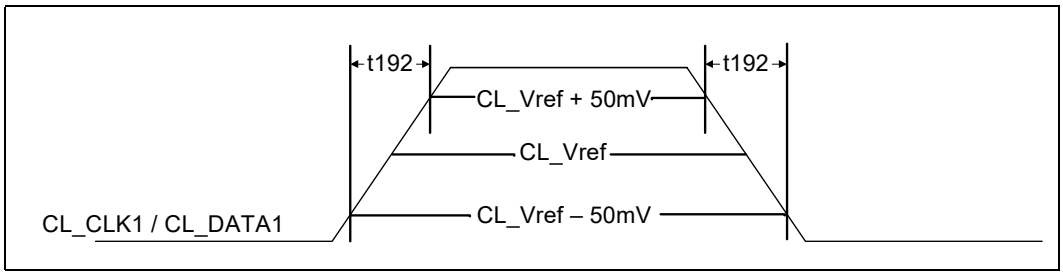
| Symbol | Parameter | Min. | Max.. | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| t190 | Single bit time | 13 | — | ns | | 10-22 |
| t191 | Single clock period | 30 | — | ns | | 10-22 |
| t193 | Setup time before CL_CLK | 0.9 | — | ns | | 10-22 |
| t194 | Hold time after CL_CLK | 0.9 | — | ns | | 10-22 |
| $V_{IL_AC}$ | Input low voltage (AC) | — | CL_Vref – 0.08 | V | 2 | |
| $V_{IH_AC}$ | Input high voltage (AC) | CL_Vref +0.08 | — | V | 2 | |

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

### Figure 10-22.Controller Link Receive Timings




### Figure 10-23.Controller Link Receive Slew Rate




### Table 10-27. UART Timings

| Sym | Parameter | Min. | Max. | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| | | | | | | |
| F | Operating Frequency | - | 6.25 | MHz | | |
| Slew_rise | Output Rise Slope | 1.452 | 2.388 | V/ns | | |
| Slew_fall | Output Fall Slope | 1.552 | 2.531 | V/ns | | |

### Table 10-28. $I^2S$ Timings (Sheet 1 of 2)

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

### Table 10-28. I²S Timings (Sheet 2 of 2)

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

## 10.6 Overshoot/Undershoot Guidelines

Overshoot (or undershoot) is the absolute value of the maximum voltage above VCC or below VSS. The PCH can be damaged by single and/or repeated overshoot or undershoot events on any input, output, or I/O buffer if the charge is large enough. Baseboard designs that meet signal integrity and timing requirements and that do not exceed the maximum overshoot or undershoot limits listed in Table 10-29 and Table 10-30 will ensure reliable I/O performance for the lifetime of the PCH.

### Table 10-29. 3.3V Overshoot/Undershoot Specifications

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

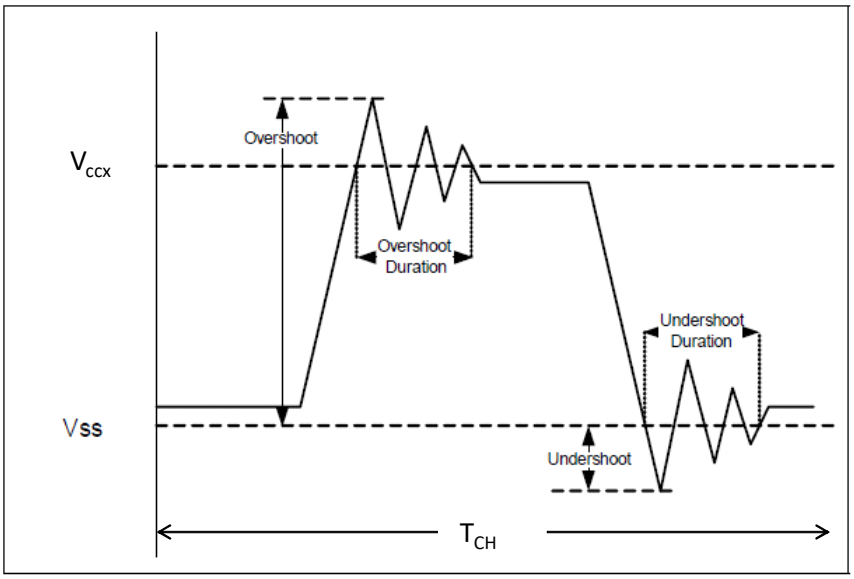
***Notes:***
1. These specifications are measured at the PCH pin.
2. Vccx refers to the supply voltage at the pin. TCH refers to the duration of the signal waveform. Refer to Figure 10-24 for pictorial description of allowable overshoot/undershoot magnitude and duration.

### Table 10-30. 1.8V Overshoot/Undershoot Specifications

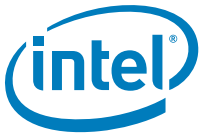
| Buffer Type | Associated Signal Group | Maximum Overshoot | Overshoot Duration | Maximum Undershoot | Undershoot Duration | Notes |
|---|---|---|---|---|---|---|
| CFIO HSHV | GPPD, GPPE | 2.57*$V_{ccx}$ | 0.25*$T_{CH}$ | -1.57*$V_{ccx}$ | 0.25*$T_{CH}$ | 1, 2 |
| CFIO I2C | GPPB, GPPC, GPPD, GPPE,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

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

### Figure 10-24. Maximum Acceptable Overshoot/Undershoot Waveform




§ §



# 11 Ballout Definition

This chapter contains the PCH Ballout information. Figure 11-1, Figure 11-2, Figure 11-3, and Figure 11-4 show the BGA ballout from a top of package quadrant view. Table 11-1 is a ball list, sorted alphabetically by signal name.

***Note:*** Ball names FAN_PWM_[3:0] and FAN_TACH_[7:0] are not used. These signals are Reserved on Desktop/Mobile SKUs.

## Figure 11-1. BGA PCH-H Ballout (Top View – Upper Left)

| | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | | VSS | VCCAMPHY PLL_1p0 | VSS | PCIE16 _TXN / SATA3_ TXN | PCIE16 _TXP / SATA3_ TXP | PCIE15 _TXP / SATA2_ TXP | | VSS | | PCIE12 _TXP | | | VSS | | DMI_T XP3 | | DMI_TXP 1 | | | VSS | | PCIE6_T XP |
| B | VSS | VSS | VCCAMPHY PLL_1p0 | | | VSS | PCIE15 _TXN / SATA2_ TXN | PCIE14 _TXN / SATA1B _TXN | VSS | PCIE13 _TXP / SATA0B _TXP | PCIE12 _TXN | | PCIE11 _TXP | PCIE10 _TXP / SATA1A _TXP | PCIE9_ TXP / SATA0A _TXP | DMI_T XN3 | DMI_TX N2 | DMI_TXN 1 | DMI_T XP0 | | VSS | PCIE8_ TXP | PCIE7_T XP |
| C | VCCMIPI PLL_1P0 | VCCMIPIP LL_1P0 | | VSS | | | | PCIE14 _TXP / SATA1B _TXP | VSS | PCIE13 _TXN / SATA0B _TXN | | | PCIE11 _TXN | PCIE10 _TXN / SATA1A _TXN | PCIE9_ TXN / SATA0A _TXN | | DMI_TX P2 | VSS | DMI_T XN0 | | | PCIE8_ TXN | PCIE7_T XN |
| D | VSS | | PCIE16_RX N / SATA3_RX N | | | | PCIE14 _RXN / SATA1B _RXN | | | VSS | VSS | | VSS | | VSS | VSS | VSS | | VSS | | VSS | VSS | |
| E | PCIE17_ TXN / SATA4_T XN | | | PCIE16 _RXP / SATA3_ RXP | PCIE15 _RXP / SATA2_ RXP | | | | PCIE14 _RXP / SATA1B _RXP | | PCIE13 _RXP / SATA0B _RXP | | VSS | | VSS | | PCIE10 _RXP / SATA1A _RXP | | | DMI_RX P2 | | DMI_R XN1 | |
| F | PCIE17_ TXP / SATA4_T XP | VSS | | | PCIE15 _RXN / SATA2_ RXN | | | | | | | | | | | | | | | | | | |
| G | PCIE18_ TXN / SATA5_T XN | PCIE18_T XP / SATA5_TX P | | VSS | | | | | PCIE18 _RXP / SATA5_ RXP | | PCIE13 _RXN / SATA0B _RXN | | PCIE12 _RXN | | PCIE9_ RXN / SATA0A _RXN | | PCIE10 _RXN / SATA1A _RXN | | DMI_R XN2 | | | DMI_R XP1 | |
| H | | PCIE19_T XP / SATA6_TX P | PCIE19_TX N / SATA6_TXN | PCIE17 _RXN / SATA4_ RXN | | PCIE17 _RXP / SATA4_ RXP | | | | | VSS | | PCIE12 _RXP | | PCIE9_ RXP / SATA0A _RXP | | VSS | | VSS | | | VSS | |
| J | PCIE20_ TXP / SATA7_T XP | | | | | | VSS | | | | | | | | | | | | | | | | |
| K | | PCIE20_T XN / SATA7_TX N | VSS | VSS | | | | | PCIE18 _RXN / SATA5_ RXN | VSS | | | VSS | | PCIE11 _RXP | | DMI_RX P3 | | VSS | | | PCIE8_ RXN | |
| L | | GPP_G22 | GPP_G23 | | VSS | | PCIE19 _RXP / SATA6_ RXP | | PCIE19 _RXN / SATA6_ RXN | | | | | | PCIE11 _RXN | | DMI_RX N3 | | DMI_R XN0 | | | PCIE8_ RXP | |
| M | GPP_G19 / SMI# | GPP_G3 / FAN_TACH _3 | | VSS | | | | | | | VSS | | | | | | | | | | | | |
| N | | GPP_G17 / ADR_COM PLETE | GPP_G18 / NMI# | GPP_G1 1 / FAN_P WM_3 | VSS | | PCIE20 _RXN / SATA7_ RXN | PCIE20 _RXP / SATA7_ RXP | | VSS | VSS | | | | RSVD | | RSVD | | DMI_R XP0 | | | VSS | |
| P | VSS | GPP_G5 / FAN_TACH _5 | GPP_G16 / GSXCLK | | | | | | | | | | | | RSVD | | RSVD | | RSVD | | | RSVD | |
| R | | GPP_G8 / FAN_PWM _0 | GPP_G9 / FAN_PWM_ 1 | GPP_G1 4 / GSXDIN | GPP_G1 5 / GSXSR ESET# | | GPP_G1 2 / GSXDO UT | VSS | | GPP_G1 3 / GSXSLO AD | GPP_G2 0 | | VSS | | | | VSS | | RSVD | | | RSVD | |
| T | GPP_G6 / FAN_TAC H_6 | GPP_G7 / FAN_TACH _7 | | VSS | | | | | | | | | | | | | | | | | | | |
| U | | | GPP_G0 / FAN_TACH_ 0 | GPP_G1 / FAN_TA CH_1 | GPP_G2 / FAN_TA CH_2 | | GPP_G1 0 / FAN_P WM_2 | VSS | | GPP_G4 / FAN_TA CH_4 | GPP_G2 1 | | VSS | VSS | VSS | | VSS | VSS | | VCCMPH Y_1p0 | VCCMPH Y_1p0 | | VCCMPH Y_1p0 |
| V | VSS | GPP_F23 | | | | | | | | | | | | | | | VSS | VCCAPLL EBB_1p0 | | VCCMPH Y_1p0 | VSS | | VSS |
| W | | GPP_F17 / USB2_OC 6# | GPP_F18 / USB2_OC7 # | GPP_F1 9 / eDP_VD DEN | | | GPP_F2 2 | VSS | | GPP_F2 1 / eDP_BK LTCTL | GPP_F2 0 / eDP_BK LTEN | | VSS | VSS | VSS | | | | | | | | |
| Y | | GPP_F14 | GPP_F15 / USB2_OC4 # | | GPP_F1 6 / USB2_ OC5# | | | | | | | | | | | | VSS | VSS | | VSS | VCCPRI M_1p0 | | VCCPRI M_1p0 |
| AA | GPP_F12 / SDATAO UT1 | GPP_F13 / SDATAOU T0 | | VSS | | | | | | | | | | | | | VSS | VCCPRIM _1p0 | | VCCPRI M_1p0 | VSS | | VCCPRI M_1p0 |
| AB | | GPP_F4 / SATAXPCI E7 / SATAGP7 | GPP_F7 / DEVSLP5 | GPP_F6 / DEVSLP 4 | GPP_F5 / DEVSLP 3 | | GPP_F9 / DEVSLP 7 | VSS | | GPP_F8 / DEVSLP 6 | GPP_F1 1 / SLOAD | | GPP_F1 0 / SCLOC K | VSS | VSS | | | | | | | | |

## Figure 11-2. BGA PCH-H Ballout (Top View – Upper Right)

| | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | | PCIE4_TXP / USB3_10_TXP | | | VSS | | PCIE1_TXN / USB3_7_TXN | | USB3_4_TXN | | USB3_2_TXP | | | USB3_3_RXP | | USB3_1_RXP | XTAL24_IN | XTAL24_OUT | VSS | VSS | VSS | |
| B | PCIE6_TXN | PCIE4_TXN / USB3_10_TXN | PCIE3_TXN / USB3_9_TXN | PCIE2_TXN / USB3_8_TXN | PCIE_RCOMPN | | PCIE1_TXP / USB3_7_TXP | USB3_6_TXN | USB3_5_TXN | USB3_4_TXP | USB3_2_TXN | USB3_1_TXP | USB3_3_RXN | | USB3_2_RXP | USB3_1_RXN | VSS | | | VSS | VSS | VSS |
| C | PCIE5_TXP | | PCIE3_TXP / USB3_9_TXP | PCIE2_TXP / USB3_8_TXP | | PCIE_RCOMPP | | USB3_6_TXP | USB3_5_TXP | USB3_3_TXN | | USB3_1_TXN | VSS | | USB3_2_RXN | | | | VSS | | VSS | RSVD |
| D | PCIE5_TXN | VSS | | VSS | | VSS | VSS | VSS | | USB3_3_TXP | VSS | | VSS | | CLKOUT_SRC_N5 | CLKOUT_SRC_P5 | | CLKOUT_SRC_N4 | | CLKOUT_SRC_N2 | | RSVD |
| E | PCIE6_RXP | | PCIE4_RXN / USB3_10_RXN | | | PCIE2_RXN / USB3_8_RXN | | VSS | | VSS | | USB3_4_RXN | | | | | CLKOUT_SRC_P4 | CLKOUT_SRC_N3 | | | | XCLK_BIASREF |
| F | | | | | | | | | | | | | | | VSS | | | | | | CLKOUT_SRC_P2 | CLKOUT_CPUNSSC_N |
| G | PCIE6_RXN | | | PCIE4_RXP / USB3_10_RXP | | PCIE2_RXP / USB3_8_RXP | | PCIE1_RXP / USB3_7_RXP | | USB3_5_RXN | | USB3_4_RXP | | VSS | | | | | CLKOUT_SRC_P3 | | CLKOUT_CPUBCLK_P | CLKOUT_CPUNSSC_P |
| H | VSS | | | VSS | | VSS | | PCIE1_RXN / USB3_7_RXN | | USB3_5_RXP | | | | | | | | | | VSS | CLKOUT_CPUBCLK_N | |
| J | | | | | | | | | | | | VSS | VSS | | | VSS | | VSS | | VSS | CLKOUT_CPUPCIBCLK_P | CLKOUT_CPUPCIBCLK_N |
| K | PCIE7_RXP | | | PCIE5_RXN | | PCIE3_RXP / USB3_9_RXP | | USB3_6_RXN | | USB3_6_RXP | | | VSS | | | | | | VSS | VCCCLK5 | VCCCLK5 | |
| L | PCIE7_RXN | | | PCIE5_RXP | | PCIE3_RXN / USB3_9_RXN | | VSS | | VSS | VSS | | | | VSS | CLKOUT_SRC_N1 | | CLKOUT_SRC_P1 | VSS | | CLKOUT_ITPXDP_P | CLKOUT_ITPXDP_N |
| M | | | | | | | | | | | | | | | | | | | | | | |
| N | VSS | | | VSS | | VCCCLK1 | | VSS | | | | | VSS | | CLKOUT_SRC_P0 | CLKOUT_SRC_N0 | | VSS | VSS | CLKOUT_SRC_N9 | CLKOUT_SRC_P9 | |
| P | VSS | | | VSS | | VSS | | | | | | | | | | | | | | CLKOUT_SRC_N10 | CLKOUT_SRC_P10 | CLKOUT_SRC_N14 |
| R | VSS | | | VCCCLK3 | | VCCCLK2 | | | VSS | CLKOUT_SRC_N15 | | CLKOUT_SRC_P15 | VSS | | CLKOUT_SRC_N6 | CLKOUT_SRC_P6 | | VSS | CLKOUT_SRC_P11 | CLKOUT_SRC_N11 | CLKOUT_SRC_P14 | |
| T | | | | | | | | | | | | | | | | | | | VSS | | VSS | VSS |
| U | | VCCMPHY_1p0 | VCCCLK4 | | VSS | VSS | | VSS | VSS | RSVD | | VSS | VSS | | VSS | CLKOUT_SRC_P7 | | CLKOUT_SRC_N7 | VSS | CLKOUT_SRC_P12 | CLKOUT_SRC_N12 | |
| V | | VSS | VSS | | VSS | VCCCLK2 | | | | | | | | | | | | | | VSS | USB2N_13 | USB2P_13 |
| W | | | | | | | | VCCDSW_3p3 | VSS | RSVD | | CLKOUT_SRC_P8 | CLKOUT_SRC_N8 | | VSS | CLKOUT_SRC_N13 | | | VSS | USB2P_11 | USB2N_11 | |
| Y | | VSS | VSS | | VSS | VSS | | | | | | | | | | | | CLKOUT_SRC_P13 | | | | |
| AA | | VSS | VSS | | VSS | VSS | | | | | | | | | | | | | VSS | | USB2P_9 | USB2N_9 |
| AB | | | | | | | | VSS | VSS | RSVD | | VSS | VSS | | VSS | VSS | | VSS | VSS | USB2N_7 | USB2P_7 | |

## Figure 11-3. BGA PCH-H Ballout (Top View – Lower Left)

| | 45 | 44 | 43 | 42 | 41 | 40 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AC | VSS | GPP_E12 / USB2_OC3 # | GPP_F3 / SATAXPCIE6 / SATAGP6 | | | | | | | | | | | | | | VSS | VCCPRIM_1p0 | | VCCPRIM_1p0 | VSS | | VCCPRIM_1p0 |
| AD | | GPP_E8 / SATALED# | GPP_E9 / USB2_OC0 # | GPP_E10 / USB2_OC1# | VCCPGPPG | | GPP_E11 / USB2_OC2# | GPP_F2 / SATAXPCIE5 / SATAGP5 | | VSS | GPP_F0 / SATAXPCIE3 / SATAGP3 | | VSS | VSS | GPP_F1 / SATAXPCIE4 / SATAGP4 | | | | | | | | |
| AE | GPP_E6 / DEVSLP2 | GPP_E7 / CPU_GP1 | | VSS | | | | | | | | | | | | | VSS | VSS | | VCCPRIM_1p0 | VSS | | VCCPRIM_1p0 |
| AF | | | | | GPP_E3 / CPU_GP0 | | | | | | | | | | | | VSS | VSS | | VSS | VSS | | VSS |
| AG | | GPP_D21 | GPP_E5 / DEVSLP1 | GPP_E4 / DEVSLP0 | | | GPP_E2 / SATAXPCIE2 / SATAGP2 | VSS | | GPP_E0 / SATAXPCIE0 / SATAGP0 | GPP_E1 / SATAXPCIE1 / SATAGP1 | | VSS | VSS | VSS | | | | | | | | |
| AH | VSS | GPP_D20 / DMIC_DATA0 | GPP_D22 | | | | | | | | | | | | | | VSS | VSS | | VSS | VSS | | VSS |
| AJ | | GPP_D23 / ISH_I2C2_SCL / I2C3_SCL | GPP_D16 / ISH_UART0_CTS# | GPP_D17 / DMIC_CLK1 | VCCPGPPEF | | GPP_D12 | GPP_D18 / DMIC_DATA1 | | VSS | GPP_D19 / DMIC_CLK0 | | GPP_D5 / I2S0_SFRM | VSS | VSS | | VSS | VSS | | VSS | VCCPRIM_1p0 | | VCCPRIM_1p0 |
| AK | GPP_D13 / ISH_UART0_RXD / SML0BDATA / I2C2_SDA | GPP_D14 / ISH_UART0_TXD / SML0BCLK / I2C2_SCL | | VSS | | | | | | | | | | | | | | | | | | | |
| AL | | GPP_D9 | GPP_D15 / ISH_UART0_RTS# | GPP_D8 / I2S0_SCLK | VCCPGPPEF | | GPP_D0 | VSS | | GPP_D10 | GPP_D11 | | VSS | VSS | | | VSS | | GPP_B1 | | | VSS | |
| AM | VSS | GPP_D4 / ISH_I2C2_SDA / I2C3_SDA | GPP_D6 / I2S0_TXD | | | | | | | | | | | | | | VSS | | VSS | | | VSS | |
| AN | | GPP_C22 / UART2_RTS# | GPP_C23 / UART2_CTS# | GPP_D7 / I2S0_RXD | GPP_D3 | | VSS | GPP_D2 | | GPP_D1 | | | | | VSS | | RSVD | | VSS | | | GPP_B11 | |
| AP | | | | | | | | | | | | | | | | | | | | | | | |
| AR | GPP_C20 / UART2_RXD | GPP_C18 / I2C1_SDA | | VSS | GPP_C19 / I2C1_SCL | | GPP_C21 / UART2_TXD | GPP_C17 / I2C0_SCL | | | | VSS | VSS | | GPP_H1 / SRCCLKREQ7# | | GPP_B21 / GSPI1_MISO | | GPP_B0 | | | GPP_B15 / GSPI0_CS# | |
| AT | | GPP_C14 / UART1_RTS# / ISH_UART1_RTS# | GPP_C13 / UART1_TXD / ISH_UART1_TXD | GPP_C16 / I2C0_SDA | | | | | | VSS | | | GPP_H0 / SRCCLKREQ6# | | SPI0_CS2# | | GPP_B22 / GSPI1_MOSI | | GPP_B23 / SML1ALERT# / PCHHOT# | | | GPP_B7 / SRCCLKREQ2# | |
| AU | VSS | GPP_C11 / UART0_CTS# | GPP_C12 / UART1_RXD / ISH_UART1_RXD | | GPP_C15 / UART1_CTS# / ISH_UART1_CTS# | | VSS | | | VSS | VSS | | | | | | | | | | | | |
| AV | | GPP_C9 / UART0_TXD | GPP_C10 / UART0_RTS# | | | | | | | | | | VSS | | VSS | | GPP_B20 / GSPI1_CLK | | VSS | | | VSS | |
| AW | GPP_C7 / SML1DATA | GPP_C0 / SMBCLK | | GPP_C6 / SML1CLK | | | | | VSS | | GPP_H11 / SML2DATA | | GPP_H7 / SRCCLKREQ13# | | SPI0_CS1# | | VSS | | GPP_B16 / GSPI0_CLK | | | GPP_B6 / SRCCLKREQ1# | |
| AY | VSS | GPP_C3 / SML0CLK | | | | | | VSS | | | | | | | | | | | | | | | |
| BA | VCCPGPPD | | | | GPP_C8 / UART0_RXD | GPP_C5 / SML0ALERT# | | | | | GPP_H14 / SML3DATA | | GPP_H6 / SRCCLKREQ12# | | VCCPGPPA | | DCPDSW_1p0 | | | DCPRTC | | VCCDSW_3p3 | |
| BB | VCCPGPPD | | GPP_C1 / SMBDATA | | GPP_C2 / SMBALERT# | | GPP_C4 / SML0DATA | GPP_H19 / ISH_I2C0_SDA | | GPP_H15 / SML3ALERT# | | VSS | GPP_H8 / SRCCLKREQ14# | | GPP_H4 / SRCCLKREQ10# | VSS | SPI0_MOSI | | GPP_B13 / PLTRST# | | VSS | GPP_B9 / SRCCLKREQ4# | |
| BC | VCCPGPPD | VCCPGPPD | | VCCPGPPBCH | | | | GPP_H20 / ISH_I2C0_SCL | | GPP_H18 / SML4ALERT# | GPP_H13 / SML3CLK | | GPP_H5 / SRCCLKREQ11# | GPP_H3 / SRCCLKREQ9# | SPI0_CLK | | SPI0_IO2 | | GPP_B19 / GSPI1_CS# | GPP_B12 / SLP_S0# | | GPP_B5 / SRCCLKREQ0# | GPP_B3 / CPU_GP2 |
| BD | VSS | VSS | VSS | | | VCCPGPPBCH | GPP_H16 / SML4CLK | GPP_H22 / ISH_I2C1_SCL | | GPP_H23 | GPP_H12 / SML2ALERT# | GPP_H10 / SML2CLK | GPP_H9 / SRCCLKREQ15# | GPP_H2 / SRCCLKREQ8# | SPI0_CS0# | SPI0_IO3 | | GPP_B18 / GSPI0_MOSI | GPP_B17 / GSPI0_MISO | GPP_B14 / SPKR | GPP_B8 / SRCCLKREQ3# | GPP_B4 / CPU_GP3 | GPP_B2 / VRALERT# |
| BE | | VSS | VCCSPI | VCCSPI | VCCSPI | VSS | GPP_H21 / ISH_I2C1_SDA | | VSS | | | GPP_H17 / SML4DATA | | VSS | | SPI0_MISO | | VSS | | | GPP_B10 / SRCCLKREQ5# | | VSS |

## Figure 11-4. BGA PCH-H Ballout (Top View – Lower Right)

| | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AC | | VSS | VSS | | VSS | VCCPRIM_1p0 | | | | | | | | | | | | | | USB2P_5 | USB2N_5 | VSS |
| AD | | | | | | | | VCCPRIM_1p0 | VSS | VCCATS | | VSS | USB2_VBUSSENSE | | VSS | USB2P_2 | | USB2N_2 | VSS | USB2N_12 | USB2P_12 | |
| AE | | VSS | VSS | | VSS | RSVD | | | | | | | | | | | | | VSS | | USB2P_4 | USB2N_4 |
| AF | | VSS | VSS | | VSS | RSVD | | | | | | | | | | | | USB2N_1 | | USB2P_6 | USB2N_6 | |
| AG | | | | | | | | RSVD | RSVD | VSS | | VSS | USB2P_3 | | USB2N_3 | USB2P_1 | | | VSS | USB2_COMP | USB2_ID | |
| AH | | VSS | VSS | | VSS | VSS | | | | | | | | | | | | | | | PM_DOWN | VSS |
| AJ | | VCCPRIM_1p0 | VCCPRIM_1p0 | | VSS | VSS | | VSS | VSS | USB2P_14 | | USB2N_14 | VSS | | USB2N_10 | USB2P_10 | | VCCUSB2PLL_1p0 | PM_SYNC | THERMTRIP# | | |
| AK | | | | | | | | | | | | | | | | | | | VSS | | PLTRST_PROC# | PCH_TRIGIN |
| AL | VCCPRIM_1p0 | | | VSS | | VSS | | | | VSS | | VSS | VSS | | USB2N_8 | USB2P_8 | | VCCUSB2PLL_1p0 | VSS | PECI | PCH_TRIGOUT | |
| AM | VSS | | | VSS | | VSS | | VSS | | | | | | | | | | | | PROCPWRGD | DISPA_BCLK | DISPA_SDO |
| AN | VSS | | | VCCHDAPLL_1P0 | | TP1 | | GPD8 / SUSCLK | | | | VSS | VSS | | VSS | VSS | | VCCPRIM_3p3 | VSS | JTAG_TCK | DISPA_SDI | |
| AP | | | | | | | | | | | | VSS | | | | | | | VSS | | JTAG_TDI | JTAG_TDO |
| AR | RSVD | | | TP2 | | GPP_A16 / CLKOUT_48 | | GPD11 / LANPHYPC | | | | | | VSS | | VSS | | VSS | | JTAGX | JTAG_TMS | |
| AT | GPP_A1 / LAD0 / ESPI_IO0 | | | GPP_A3 / LAD2 / ESPI_IO2 | | GPP_A0 / RCIN# / ESPI_ALERT1# | | VSS | | GPD3 / PWRBTN# | | | VSS | VSS | | | | | PRDY# | PREQ# | ITP_PMODE | |
| AU | | | | | | | | | | | | | | | | VSS | | | | | | VSS |
| AV | GPP_A2 / LAD1 / ESPI_IO1 | | | GPP_A10 / CLKOUT_LPC1 | | VSS | | SLP_LAN# | | GPD9 / SLP_WLAN# | | DSW_PWROK | | | | | VSS | | GPP_I2 / DDPD_HPD2 | CL_DATA | CL_CLK | |
| AW | GPP_A8 / CLKRUN# | | | VSS | | GPP_A7 / PIRQA# / ESPI_ALERT0# | | GPD4 / SLP_S3# | | VSS | | PCH_PWROK | | VSS | | | | | GPP_I0 / DDPB_HPD0 | | CL_RST# | SYS_RESET# |
| AY | | | | | | | | | | | | | | | | | | CPU_TRST# | | | GPP_I1 / DDPC_HPD1 | SYS_PWROK |
| BA | VCCRTC | | VCCRTCPRIM_3p3 | | | GPP_A6 / SERIRQ / ESPI_CS1# | | VCCHDA | | GPD10 / SLP_S5# | | RSMRST# | | HDA_BCLK | | | | GPP_I5 / DDPB_CTRLCLK | GPP_I3 / DDPE_HPD3 | | | VSS |
| BB | GPP_A18 / ISH_GP0 | VSS | | GPP_A15 / SUSACK# | | GPP_A12 / BMBUSY# / ISH_GP6 / SX_EXIT_HOLDOFF# | VSS | GPD1 / ACPRESENT | | SLP_SUS# | | VSS | SRTCRST# | | | HDA_SDO | | | | GPP_I7 / DDPC_CTRLCLK | | VSS |
| BC | GPP_A23 / ISH_GP5 | | | GPP_A17 / ISH_GP7 | GPP_A14 / SUS_STAT# / ESPI_RESET# | GPP_A9 / CLKOUT_LPC0 / ESPI_CLK | | GPD6 / SLP_A# | DRAM_RESET# | WAKE# | | | RTCRST# | RTCX1 | HDA_SDI1 | | | | GPP_I6 / DDPB_CTRLDATA | | VSS | VSS |
| BD | GPP_A20 / ISH_GP2 | GPP_A19 / ISH_GP1 | | GPP_A13 / SUSWARN# / SUSPWRDNACK | GPP_A22 / ISH_GP4 | GPP_A11 / PME# | GPP_A4 / LAD3 / ESPI_IO3 | GPD5 / SLP_S4# | GPD7 / RSVD | GPD0 / BATLOW# | | GPD2 / LAN_WAKE# | RTCX2 | HDA_SYNC | HDA_RST# | GPP_I4 / EDP_HPD | GPP_I8 / DDPC_CTRLDATA | | | VCCPRIM_3p3 | VSS | RSVD |
| BE | | GPP_A21 / ISH_GP3 | | | VSS | | GPP_A5 / LFRAME# / ESPI_CS0# | | VSS | | | INTRUDER# | | VSS | | HDA_SDI0 | GPP_I10 / DDPD_CTRLDATA | GPP_I9 / DDPD_CTRLCLK | VCCPRIM_3p3 | VCCPRIM_3p3 | RSVD | |

### Table 11-1. BGA PCH-H Ballout (Sheet 1 of 20)

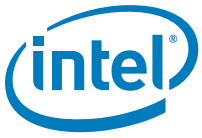
| Ball Name | Ball # |
|---|---|
| CL_CLK | AV2 |
| CL_DATA | AV3 |
| CL_RST# | AW2 |
| CLKOUT_CPUBCLK_N | H2 |
| CLKOUT_CPUBCLK_P | G2 |
| CLKOUT_CPUNSSC_N | F1 |
| CLKOUT_CPUNSSC_P | G1 |
| CLKOUT_CPUPCIBCLK_N | J1 |
| CLKOUT_CPUPCIBCLK_P | J2 |
| CLKOUT_ITPXDP_N | L1 |
| CLKOUT_ITPXDP_P | L2 |
| CLKOUT_SRC_N0 | N7 |
| CLKOUT_SRC_N1 | L7 |
| CLKOUT_SRC_N10 | P3 |
| CLKOUT_SRC_N11 | R3 |
| CLKOUT_SRC_N12 | U2 |
| CLKOUT_SRC_N13 | W7 |
| CLKOUT_SRC_N14 | P1 |
| CLKOUT_SRC_N15 | R13 |
| CLKOUT_SRC_N2 | D3 |
| CLKOUT_SRC_N3 | E5 |
| CLKOUT_SRC_N4 | D5 |
| CLKOUT_SRC_N5 | D8 |
| CLKOUT_SRC_N6 | R8 |
| CLKOUT_SRC_N7 | U5 |
| CLKOUT_SRC_N8 | W10 |
| CLKOUT_SRC_N9 | N3 |
| CLKOUT_SRC_P0 | N8 |
| CLKOUT_SRC_P1 | L5 |
| CLKOUT_SRC_P10 | P2 |
| CLKOUT_SRC_P11 | R4 |
| CLKOUT_SRC_P12 | U3 |
| CLKOUT_SRC_P13 | Y5 |
| CLKOUT_SRC_P14 | R2 |
| CLKOUT_SRC_P15 | R11 |
| CLKOUT_SRC_P2 | F2 |
| CLKOUT_SRC_P3 | G4 |
| CLKOUT_SRC_P4 | E6 |
| CLKOUT_SRC_P5 | D7 |
| CLKOUT_SRC_P6 | R7 |
| CLKOUT_SRC_P7 | U7 |
| CLKOUT_SRC_P8 | W11 |
| CLKOUT_SRC_P9 | N2 |
| CPU_TRST# | AY5 |
| DCPDSW_1p0 | BA29 |

### Table 11-1. BGA PCH-H Ballout (Sheet 2 of 20)

| Ball Name | Ball # |
|---|---|
| DCPRTC | BA26 |
| DISPA_BCLK | AM2 |
| DISPA_SDI | AN2 |
| DISPA_SDO | AM1 |
| DMI_RXN0 | L27 |
| DMI_RXN1 | E24 |
| DMI_RXN2 | G27 |
| DMI_RXN3 | L29 |
| DMI_RXP0 | N27 |
| DMI_RXP1 | G24 |
| DMI_RXP2 | E26 |
| DMI_RXP3 | K29 |
| DMI_TXN0 | C27 |
| DMI_TXN1 | B28 |
| DMI_TXN2 | B29 |
| DMI_TXN3 | B30 |
| DMI_TXP0 | B27 |
| DMI_TXP1 | A28 |
| DMI_TXP2 | C29 |
| DMI_TXP3 | A30 |
| DRAM_RESET# | BC14 |
| DSW_PWROK | AV11 |
| GPD0 / BATLOW# | BD13 |
| GPD1 / ACPRESENT | BB15 |
| GPD10 / SLP_S5# | BA13 |
| GPD11 / LANPHYPC | AR15 |
| GPD2 / LAN_WAKE# | BD11 |
| GPD3 / PWRBTN# | AT13 |
| GPD4 / SLP_S3# | AW15 |
| GPD5 / SLP_S4# | BD15 |
| GPD6 / SLP_A# | BC15 |
| GPD7 / RSVD | BD14 |
| GPD8 / SUSCLK | AN15 |
| GPD9 / SLP_WLAN# | AV13 |
| GPP_A0 / RCIN# / ESPI_ALERT1# | AT17 |
| GPP_A1 / LAD0 / ESPI_IO0 | AT22 |
| GPP_A10 / CLKOUT_LPC1 | AV19 |
| GPP_A11 /PME# | BD17 |
| GPP_A12 / BMBUSY# / ISH_GP6 / SX_EXIT_HOLDOFF# | BB17 |
| GPP_A13 / SUSWARN# / SUSPWRDNACK | BD19 |
| GPP_A14 / SUS_STAT#/ ESPI_RESET# | BC18 |

### Table 11-1. BGA PCH-H Ballout (Sheet 3 of 20)

| Ball Name | Ball # |
|---|---|
| GPP_A15 / SUSACK# | BB19 |
| GPP_A16 / CLKOUT_48 | AR17 |
| GPP_A17 / ISH_GP7 | BC19 |
| GPP_A18 / ISH_GP0 | BB22 |
| GPP_A19 / ISH_GP1 | BD21 |
| GPP_A2 / LAD1 / ESPI_IO1 | AV22 |
| GPP_A20 / ISH_GP2 | BD22 |
| GPP_A21 / ISH_GP3 | BE21 |
| GPP_A22 / ISH_GP4 | BD18 |
| GPP_A23 / ISH_GP5 | BC22 |
| GPP_A3 / LAD2 / ESPI_IO2 | AT19 |
| GPP_A4 / LAD3 / ESPI_IO3 | BD16 |
| GPP_A5 / LFRAME# / ESPI_CS0# | BE16 |
| GPP_A6 / SERIRQ / ESPI_CS1# | BA17 |
| GPP_A7 / PIRQA# / ESPI_ALERT0# | AW17 |
| GPP_A8 / CLKRUN# | AW22 |
| GPP_A9 / CLKOUT_LPC0 / ESPI_CLK | BC17 |
| GPP_B0 | AR27 |
| GPP_B1 | AL27 |
| GPP_B10 / SRCCLKREQ5# | BE25 |
| GPP_B11 | AN24 |
| GPP_B12 / SLP_S0# | BC26 |
| GPP_B13 / PLTRST# | BB27 |
| GPP_B14 / SPKR | BD26 |
| GPP_B15 / GSPI0_CS# | AR24 |
| GPP_B16 / GSPI0_CLK | AW27 |
| GPP_B17 / GSPI0_MISO | BD27 |
| GPP_B18 / GSPI0_MOSI | BD28 |
| GPP_B19 / GSPI1_CS# | BC27 |
| GPP_B2 / VRALERT# | BD23 |
| GPP_B20 / GSPI1_CLK | AV29 |
| GPP_B21 / GSPI1_MISO | AR29 |
| GPP_B22 / GSPI1_MOSI | AT29 |
| GPP_B23 / SML1ALERT# / PCHHOT# | AT27 |
| GPP_B3 / CPU_GP2 | BC23 |
| GPP_B4 / CPU_GP3 | BD24 |



### Table 11-1. BGA PCH-H Ballout (Sheet 4 of 20)

| Ball Name | Ball # |
|---|---|
| GPP_B5 / SRCCLKREQ0# | BC24 |
| GPP_B6 / SRCCLKREQ1# | AW24 |
| GPP_B7 / SRCCLKREQ2# | AT24 |
| GPP_B8 / SRCCLKREQ3# | BD25 |
| GPP_B9 / SRCCLKREQ4# | BB24 |
| GPP_C0 / SMBCLK | AW44 |
| GPP_C1 / SMBDATA | BB43 |
| GPP_C10 / UART0_RTS# | AV43 |
| GPP_C11 / UART0_CTS# | AU44 |
| GPP_C12 / UART1_RXD / ISH_UART1_RXD | AU43 |
| GPP_C13 / UART1_TXD / ISH_UART1_TXD | AT43 |
| GPP_C14 / UART1_RTS# / ISH_UART1_RTS# | AT44 |
| GPP_C15 / UART1_CTS# / ISH_UART1_CTS# | AU41 |
| GPP_C16 / I2C0_SDA | AT42 |
| GPP_C17 / I2C0_SCL | AR38 |
| GPP_C18 / I2C1_SDA | AR44 |
| GPP_C19 / I2C1_SCL | AR41 |
| GPP_C2 / SMBALERT# | BB41 |
| GPP_C20 / UART2_RXD | AR45 |
| GPP_C21 / UART2_TXD | AR39 |
| GPP_C22 / UART2_RTS# | AN44 |
| GPP_C23 / UART2_CTS# | AN43 |
| GPP_C3 / SML0CLK | AY44 |
| GPP_C4 / SML0DATA | BB39 |
| GPP_C5 / SML0ALERT# | BA40 |
| GPP_C6 / SML1CLK | AW42 |
| GPP_C7 / SML1DATA | AW45 |
| GPP_C8 / UART0_RXD | BA41 |
| GPP_C9 / UART0_TXD | AV44 |
| GPP_D0 | AL39 |
| GPP_D1 | AN36 |
| GPP_D10 | AL36 |
| GPP_D11 | AL35 |
| GPP_D12 | AJ39 |
| GPP_D13 / ISH_UART0_RXD / SML0BDATA / I2C2_SDA | AK45 |

### Table 11-1. BGA PCH-H Ballout (Sheet 5 of 20)

| Ball Name | Ball # |
|---|---|
| GPP_D14 / ISH_UART0_TXD / SML0BCLK / I2C2_SCL | AK44 |
| GPP_D15 / ISH_UART0_RTS# | AL43 |
| GPP_D16 / ISH_UART0_CTS# | AJ43 |
| GPP_D17 / DMIC_CLK1 | AJ42 |
| GPP_D18 / DMIC_DATA1 | AJ38 |
| GPP_D19 / DMIC_CLK0 | AJ35 |
| GPP_D2 | AN38 |
| GPP_D20 / DMIC_DATA0 | AH44 |
| GPP_D21 | AG44 |
| GPP_D22 | AH43 |
| GPP_D23 / ISH_I2C2_SCL / I2C3_SCL | AJ44 |
| GPP_D3 | AN41 |
| GPP_D4 / ISH_I2C2_SDA / I2C3_SDA | AM44 |
| GPP_D5 / I2S0_SFRM | AJ33 |
| GPP_D6 / I2S0_TXD | AM43 |
| GPP_D7 / I2S0_RXD | AN42 |
| GPP_D8 / I2S0_SCLK | AL42 |
| GPP_D9 | AL44 |
| GPP_E0 / SATAXPCIE0 / SATAGP0 | AG36 |
| GPP_E1 / SATAXPCIE1 / SATAGP1 | AG35 |
| GPP_E10 / USB2_OC1# | AD42 |
| GPP_E11 / USB2_OC2# | AD39 |
| GPP_E12 / USB2_OC3# | AC44 |
| GPP_E2 / SATAXPCIE2 / SATAGP2 | AG39 |
| GPP_E3 / CPU_GP0 | AF41 |
| GPP_E4 / DEVSLP0 | AG42 |
| GPP_E5 / DEVSLP1 | AG43 |
| GPP_E6 / DEVSLP2 | AE45 |
| GPP_E7 / CPU_GP1 | AE44 |
| GPP_E8 / SATALED# | AD44 |
| GPP_E9 / USB2_OC0# | AD43 |
| GPP_F0 / SATAXPCIE3 / SATAGP3 | AD35 |
| GPP_F1 / SATAXPCIE4 / SATAGP4 | AD31 |
| GPP_F10 / SCLOCK | AB33 |
| GPP_F11 / SLOAD | AB35 |
| GPP_F12 / SDATAOUT1 | AA45 |
| GPP_F13 / SDATAOUT0 | AA44 |

### Table 11-1. BGA PCH-H Ballout (Sheet 6 of 20)

| Ball Name | Ball # |
|---|---|
| GPP_F14 | Y44 |
| GPP_F15 / USB2_OC4# | Y43 |
| GPP_F16 / USB2_OC5# | Y41 |
| GPP_F17 / USB2_OC6# | W44 |
| GPP_F18 / USB2_OC7# | W43 |
| GPP_F19 / eDP_VDDEN | W42 |
| GPP_F2 / SATAXPCIE5 / SATAGP5 | AD38 |
| GPP_F20 / eDP_BKLTEN | W35 |
| GPP_F21 / eDP_BKLTCTL | W36 |
| GPP_F22 | W39 |
| GPP_F23 | V44 |
| GPP_F3 / SATAXPCIE6 / SATAGP6 | AC43 |
| GPP_F4 / SATAXPCIE7 / SATAGP7 | AB44 |
| GPP_F5 / DEVSLP3 | AB41 |
| GPP_F6 / DEVSLP4 | AB42 |
| GPP_F7 / DEVSLP5 | AB43 |
| GPP_F8 / DEVSLP6 | AB36 |
| GPP_F9 / DEVSLP7 | AB39 |
| GPP_G0 / FAN_TACH_0 | U43 |
| GPP_G1 / FAN_TACH_1 | U42 |
| GPP_G10 / FAN_PWM_2 | U39 |
| GPP_G11 / FAN_PWM_3 | N42 |
| GPP_G12 / GSXDOUT | R39 |
| GPP_G13 / GSXSLOAD | R36 |
| GPP_G14 / GSXDIN | R42 |
| GPP_G15 / GSXSRESET# | R41 |
| GPP_G16 / GSXCLK | P43 |
| GPP_G17 / ADR_COMPLETE | N44 |
| GPP_G18 / NMI# | N43 |
| GPP_G19 / SMI# | M45 |
| GPP_G2 / FAN_TACH_2 | U41 |
| GPP_G20 | R35 |
| GPP_G21 | U35 |
| GPP_G22 | L44 |
| GPP_G23 | L43 |
| GPP_G3 / FAN_TACH_3 | M44 |
| GPP_G4 / FAN_TACH_4 | U36 |
| GPP_G5 / FAN_TACH_5 | P44 |
| GPP_G6 / FAN_TACH_6 | T45 |
| GPP_G7 / FAN_TACH_7 | T44 |
| GPP_G8 / FAN_PWM_0 | R44 |

### Table 11-1. BGA PCH-H Ballout (Sheet 7 of 20)

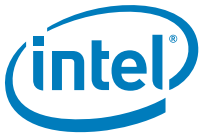
| Ball Name | Ball # |
|---|---|
| GPP_G9 / FAN_PWM_1 | R43 |
| GPP_H0 / SRCCLKREQ6# | AT33 |
| GPP_H1 / SRCCLKREQ7# | AR31 |
| GPP_H10 / SML2CLK | BD34 |
| GPP_H11 / SML2DATA | AW35 |
| GPP_H12 / SML2ALERT# | BD35 |
| GPP_H13 / SML3CLK | BC35 |
| GPP_H14 / SML3DATA | BA35 |
| GPP_H15 / SML3ALERT# | BB36 |
| GPP_H16 / SML4CLK | BD39 |
| GPP_H17 / SML4DATA | BE34 |
| GPP_H18 / SML4ALERT# | BC36 |
| GPP_H19 / ISH_I2C0_SDA | BB38 |
| GPP_H2 / SRCCLKREQ8# | BD32 |
| GPP_H20 / ISH_I2C0_SCL | BC38 |
| GPP_H21 / ISH_I2C1_SDA | BE39 |
| GPP_H22 / ISH_I2C1_SCL | BD38 |
| GPP_H23 | BD36 |
| GPP_H3 / SRCCLKREQ9# | BC32 |
| GPP_H4 / SRCCLKREQ10# | BB31 |
| GPP_H5 / SRCCLKREQ11# | BC33 |
| GPP_H6 / SRCCLKREQ12# | BA33 |
| GPP_H7 / SRCCLKREQ13# | AW33 |
| GPP_H8 / SRCCLKREQ14# | BB33 |
| GPP_H9 / SRCCLKREQ15# | BD33 |
| GPP_I0 / DDPB_HPD0 | AW4 |
| GPP_I1 / DDPC_HPD1 | AY2 |
| GPP_I10 / DDPD_CTRLDATA | BE6 |
| GPP_I2 / DDPD_HPD2 | AV4 |
| GPP_I3 / DDPE_HPD3 | BA4 |
| GPP_I4 / EDP_HPD | BD7 |
| GPP_I5 / DDPB_CTRLCLK | BA5 |
| GPP_I6 / DDPB_CTRLDATA | BC4 |

### Table 11-1. BGA PCH-H Ballout (Sheet 8 of 20)

| Ball Name | Ball # |
|---|---|
| GPP_I7 / DDPC_CTRLCLK | BB3 |
| GPP_I8 / DDPC_CTRLDATA | BD6 |
| GPP_I9 / DDPD_CTRLCLK | BE5 |
| HDA_BCLK | BA9 |
| HDA_RST# | BD8 |
| HDA_SDI0 | BE7 |
| HDA_SDI1 | BC8 |
| HDA_SDO | BB7 |
| HDA_SYNC | BD9 |
| INTRUDER# | BE11 |
| ITP_PMODE | AT2 |
| JTAG_TCK | AN3 |
| JTAG_TDI | AP2 |
| JTAG_TDO | AP1 |
| JTAG_TMS | AR2 |
| JTAGX | AR3 |
| PCH_PWROK | AW11 |
| PCH_TRIGIN | AK1 |
| PCH_TRIGOUT | AL2 |
| PCIE_RCOMPN | B18 |
| PCIE_RCOMPP | C17 |
| PCIE1_RXN / USB3_7_RXN | H15 |
| PCIE1_RXP / USB3_7_RXP | G15 |
| PCIE1_TXN / USB3_7_TXN | A16 |
| PCIE1_TXP / USB3_7_TXP | B16 |
| PCIE10_RXN / SATA1A_RXN | G29 |
| PCIE10_RXP / SATA1A_RXP | E29 |
| PCIE10_TXN / SATA1A_TXN | C32 |
| PCIE10_TXP / SATA1A_TXP | B32 |
| PCIE11_RXN | L31 |
| PCIE11_RXP | K31 |
| PCIE11_TXN | C33 |
| PCIE11_TXP | B33 |
| PCIE12_RXN | G33 |
| PCIE12_RXP | H33 |
| PCIE12_TXN | B35 |
| PCIE12_TXP | A35 |
| PCIE13_RXN / SATA0B_RXN | G35 |

### Table 11-1. BGA PCH-H Ballout (Sheet 9 of 20)

| Ball Name | Ball # |
|---|---|
| PCIE13_RXP / SATA0B_RXP | E35 |
| PCIE13_TXN / SATA0B_TXN | C36 |
| PCIE13_TXP / SATA0B_TXP | B36 |
| PCIE14_RXN / SATA1B_RXN | D39 |
| PCIE14_RXP / SATA1B_RXP | E37 |
| PCIE14_TXN / SATA1B_TXN | B38 |
| PCIE14_TXP / SATA1B_TXP | C38 |
| PCIE15_RXN / SATA2_RXN | F41 |
| PCIE15_RXP / SATA2_RXP | E41 |
| PCIE15_TXN / SATA2_TXN | B39 |
| PCIE15_TXP / SATA2_TXP | A39 |
| PCIE16_RXN / SATA3_RXN | D43 |
| PCIE16_RXP / SATA3_RXP | E42 |
| PCIE16_TXN / SATA3_TXN | A41 |
| PCIE16_TXP / SATA3_TXP | A40 |
| PCIE17_RXN / SATA4_RXN | H42 |
| PCIE17_RXP / SATA4_RXP | H40 |
| PCIE17_TXN / SATA4_TXN | E45 |
| PCIE17_TXP / SATA4_TXP | F45 |
| PCIE18_RXN / SATA5_RXN | K37 |
| PCIE18_RXP / SATA5_RXP | G37 |
| PCIE18_TXN / SATA5_TXN | G45 |
| PCIE18_TXP / SATA5_TXP | G44 |
| PCIE19_RXN / SATA6_RXN | L37 |
| PCIE19_RXP / SATA6_RXP | L39 |
| PCIE19_TXN / SATA6_TXN | H43 |
| PCIE19_TXP / SATA6_TXP | H44 |
| PCIE2_RXN / USB3_8_RXN | E17 |



### Table 11-1. BGA PCH-H Ballout (Sheet 10 of 20)

| Ball Name | Ball # |
|---|---|
| PCIE2_RXP / USB3_8_RXP | G17 |
| PCIE2_TXN / USB3_8_TXN | B19 |
| PCIE2_TXP / USB3_8_TXP | C19 |
| PCIE20_RXN / SATA7_RXN | N39 |
| PCIE20_RXP / SATA7_RXP | N38 |
| PCIE20_TXN / SATA7_TXN | K44 |
| PCIE20_TXP / SATA7_TXP | J45 |
| PCIE3_RXN / USB3_9_RXN | L17 |
| PCIE3_RXP / USB3_9_RXP | K17 |
| PCIE3_TXN / USB3_9_TXN | B20 |
| PCIE3_TXP / USB3_9_TXP | C20 |
| PCIE4_RXN / USB3_10_RXN | E20 |
| PCIE4_RXP / USB3_10_RXP | G19 |
| PCIE4_TXN / USB3_10_TXN | B21 |
| PCIE4_TXP / USB3_10_TXP | A21 |
| PCIE5_RXN | K19 |
| PCIE5_RXP | L19 |
| PCIE5_TXN | D22 |
| PCIE5_TXP | C22 |
| PCIE6_RXN | G22 |
| PCIE6_RXP | E22 |
| PCIE6_TXN | B22 |
| PCIE6_TXP | A23 |
| PCIE7_RXN | L22 |
| PCIE7_RXP | K22 |
| PCIE7_TXN | C23 |
| PCIE7_TXP | B23 |
| PCIE8_RXN | K24 |
| PCIE8_RXP | L24 |
| PCIE8_TXN | C24 |
| PCIE8_TXP | B24 |
| PCIE9_RXN / SATA0A_RXN | G31 |
| PCIE9_RXP / SATA0A_RXP | H31 |
| PCIE9_TXN / SATA0A_TXN | C31 |

### Table 11-1. BGA PCH-H Ballout (Sheet 11 of 20)

| Ball Name | Ball # |
|---|---|
| PCIE9_TXP / SATA0A_TXP | B31 |
| PECI | AL3 |
| PLTRST_PROC# | AK2 |
| PM_DOWN | AH2 |
| PM_SYNC | AJ4 |
| PRDY# | AT4 |
| PREQ# | AT3 |
| PROCPWRGD | AM3 |
| RSMRST# | BA11 |
| RSVD | C1 |
| RSVD | D1 |
| RSVD | N29 |
| RSVD | N31 |
| RSVD | P24 |
| RSVD | P27 |
| RSVD | P29 |
| RSVD | P31 |
| RSVD | R24 |
| RSVD | R27 |
| RSVD | U13 |
| RSVD | W13 |
| RSVD | AB13 |
| RSVD | AE17 |
| RSVD | AF17 |
| RSVD | AG14 |
| RSVD | AG15 |
| RSVD | AN29 |
| RSVD | AR22 |
| RSVD | BD1 |
| RSVD | BE2 |
| RTCRST# | BC10 |
| RTCX1 | BC9 |
| RTCX2 | BD10 |
| SLP_LAN# | AV15 |
| SLP_SUS# | BB13 |
| SPI0_CLK | BC31 |
| SPI0_CS0# | BD31 |
| SPI0_CS1# | AW31 |
| SPI0_CS2# | AT31 |
| SPI0_IO2 | BC29 |
| SPI0_IO3 | BD30 |
| SPI0_MISO | BE30 |
| SPI0_MOSI | BB29 |
| SRTCRST# | BB10 |
| SYS_PWROK | AY1 |
| SYS_RESET# | AW1 |

### Table 11-1. BGA PCH-H Ballout (Sheet 12 of 20)

| Ball Name | Ball # |
|---|---|
| THERMTRIP# | AJ3 |
| TP1 | AN17 |
| TP2 | AR19 |
| USB2_COMP | AG3 |
| USB2_ID | AG2 |
| USB2_VBUSSENSE | AD10 |
| USB2N_1 | AF5 |
| USB2N_10 | AJ8 |
| USB2N_11 | W2 |
| USB2N_12 | AD3 |
| USB2N_13 | V2 |
| USB2N_14 | AJ11 |
| USB2N_2 | AD5 |
| USB2N_3 | AG8 |
| USB2N_4 | AE1 |
| USB2N_5 | AC2 |
| USB2N_6 | AF2 |
| USB2N_7 | AB3 |
| USB2N_8 | AL8 |
| USB2N_9 | AA1 |
| USB2P_1 | AG7 |
| USB2P_10 | AJ7 |
| USB2P_11 | W3 |
| USB2P_12 | AD2 |
| USB2P_13 | V1 |
| USB2P_14 | AJ13 |
| USB2P_2 | AD7 |
| USB2P_3 | AG10 |
| USB2P_4 | AE2 |
| USB2P_5 | AC3 |
| USB2P_6 | AF3 |
| USB2P_7 | AB2 |
| USB2P_8 | AL7 |
| USB2P_9 | AA2 |
| USB3_1_RXN | B7 |
| USB3_1_RXP | A7 |
| USB3_1_TXN | C11 |
| USB3_1_TXP | B11 |
| USB3_2_RXN | C8 |
| USB3_2_RXP | B8 |
| USB3_2_TXN | B12 |
| USB3_2_TXP | A12 |
| USB3_3_RXN | B10 |
| USB3_3_RXP | A9 |
| USB3_3_TXN | C13 |
| USB3_3_TXP | D13 |
| USB3_4_RXN | E11 |

### Table 11-1. BGA PCH-H Ballout (Sheet 13 of 20)

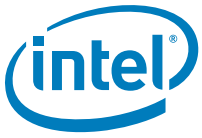
| Ball Name | Ball # |
|---|---|
| USB3_4_RXP | G11 |
| USB3_4_TXN | A14 |
| USB3_4_TXP | B13 |
| USB3_5_RXN | G13 |
| USB3_5_RXP | H13 |
| USB3_5_TXN | B14 |
| USB3_5_TXP | C14 |
| USB3_6_RXN | K15 |
| USB3_6_RXP | K13 |
| USB3_6_TXN | B15 |
| USB3_6_TXP | C15 |
| VCCAMPHYPLL_1p0 | A43 |
| VCCAMPHYPLL_1p0 | B43 |
| VCCAPLLEBB_1p0 | V28 |
| VCCATS | AD13 |
| VCCCLK1 | N17 |
| VCCCLK2 | R17 |
| VCCCLK2 | V17 |
| VCCCLK3 | R19 |
| VCCCLK4 | U20 |
| VCCCLK5 | K2 |
| VCCCLK5 | K3 |
| VCCDSW_3p3 | W15 |
| VCCDSW_3p3 | BA24 |
| VCCHDA | BA15 |
| VCCHDAPLL_1P0 | AN19 |
| VCCMPHY_1p0 | U21 |
| VCCMPHY_1p0 | U23 |
| VCCMPHY_1p0 | U25 |
| VCCMPHY_1p0 | U26 |
| VCCMPHY_1p0 | V26 |
| VCCMIPIPLL_1P0 | C44 |
| VCCMIPIPLL_1P0 | C45 |
| VCCPGPPA | BA31 |
| VCCPGPPBCH | BC42 |
| VCCPGPPBCH | BD40 |
| VCCPGPPD | BA45 |
| VCCPGPPD | BB45 |
| VCCPGPPD | BC44 |
| VCCPGPPD | BC45 |
| VCCPGPPEF | AJ41 |
| VCCPGPPEF | AL41 |
| VCCPGPPG | AD41 |
| VCCPRIM_1p0 | Y23 |
| VCCPRIM_1p0 | Y25 |
| VCCPRIM_1p0 | AA23 |
| VCCPRIM_1p0 | AA26 |

### Table 11-1. BGA PCH-H Ballout (Sheet 14 of 20)

| Ball Name | Ball # |
|---|---|
| VCCPRIM_1p0 | AA28 |
| VCCPRIM_1p0 | AC17 |
| VCCPRIM_1p0 | AC23 |
| VCCPRIM_1p0 | AC26 |
| VCCPRIM_1p0 | AC28 |
| VCCPRIM_1p0 | AD15 |
| VCCPRIM_1p0 | AE23 |
| VCCPRIM_1p0 | AE26 |
| VCCPRIM_1p0 | AJ20 |
| VCCPRIM_1p0 | AJ21 |
| VCCPRIM_1p0 | AJ23 |
| VCCPRIM_1p0 | AJ25 |
| VCCPRIM_1p0 | AL22 |
| VCCPRIM_3p3 | AN5 |
| VCCPRIM_3p3 | BD3 |
| VCCPRIM_3p3 | BE3 |
| VCCPRIM_3p3 | BE4 |
| VCCRTC | BA22 |
| VCCRTCPRIM_3p3 | BA20 |
| VCCSPI | BE41 |
| VCCSPI | BE42 |
| VCCSPI | BE43 |
| VCCUSB2PLL_1p0 | AJ5 |
| VCCUSB2PLL_1p0 | AL5 |
| VSS | A2 |
| VSS | A3 |
| VSS | A4 |
| VSS | A18 |
| VSS | A25 |
| VSS | A32 |
| VSS | A37 |
| VSS | A42 |
| VSS | A44 |
| VSS | B1 |
| VSS | B2 |
| VSS | B3 |
| VSS | B6 |
| VSS | B25 |
| VSS | B37 |
| VSS | B40 |
| VSS | B44 |
| VSS | B45 |
| VSS | C2 |
| VSS | C4 |
| VSS | C10 |
| VSS | C28 |
| VSS | C37 |

### Table 11-1. BGA PCH-H Ballout (Sheet 15 of 20)

| Ball Name | Ball # |
|---|---|
| VSS | C42 |
| VSS | D10 |
| VSS | D12 |
| VSS | D15 |
| VSS | D16 |
| VSS | D17 |
| VSS | D19 |
| VSS | D21 |
| VSS | D24 |
| VSS | D25 |
| VSS | D27 |
| VSS | D29 |
| VSS | D30 |
| VSS | D31 |
| VSS | D33 |
| VSS | D35 |
| VSS | D36 |
| VSS | D45 |
| VSS | E13 |
| VSS | E15 |
| VSS | E31 |
| VSS | E33 |
| VSS | F8 |
| VSS | F44 |
| VSS | G9 |
| VSS | G42 |
| VSS | H3 |
| VSS | H17 |
| VSS | H19 |
| VSS | H22 |
| VSS | H24 |
| VSS | H27 |
| VSS | H29 |
| VSS | H35 |
| VSS | J3 |
| VSS | J5 |
| VSS | J7 |
| VSS | J10 |
| VSS | J11 |
| VSS | J39 |
| VSS | K4 |
| VSS | K10 |
| VSS | K27 |
| VSS | K33 |
| VSS | K36 |
| VSS | K42 |
| VSS | K43 |



### Table 11-1. BGA PCH-H Ballout (Sheet 16 of 20)

| Ball Name | Ball # |
|---|---|
| VSS | L4 |
| VSS | L8 |
| VSS | L12 |
| VSS | L13 |
| VSS | L15 |
| VSS | L41 |
| VSS | M35 |
| VSS | M42 |
| VSS | N4 |
| VSS | N5 |
| VSS | N10 |
| VSS | N15 |
| VSS | N19 |
| VSS | N22 |
| VSS | N24 |
| VSS | N35 |
| VSS | N36 |
| VSS | N41 |
| VSS | P17 |
| VSS | P19 |
| VSS | P22 |
| VSS | P45 |
| VSS | R5 |
| VSS | R10 |
| VSS | R14 |
| VSS | R22 |
| VSS | R29 |
| VSS | R33 |
| VSS | R38 |
| VSS | T1 |
| VSS | T2 |
| VSS | T4 |
| VSS | T42 |
| VSS | U4 |
| VSS | U8 |
| VSS | U10 |
| VSS | U11 |
| VSS | U14 |
| VSS | U15 |
| VSS | U17 |
| VSS | U18 |
| VSS | U28 |
| VSS | U29 |
| VSS | U31 |
| VSS | U32 |
| VSS | U33 |
| VSS | U38 |

### Table 11-1. BGA PCH-H Ballout (Sheet 17 of 20)

| Ball Name | Ball # |
|---|---|
| VSS | V3 |
| VSS | V18 |
| VSS | V20 |
| VSS | V21 |
| VSS | V23 |
| VSS | V25 |
| VSS | V29 |
| VSS | V45 |
| VSS | W4 |
| VSS | W8 |
| VSS | W14 |
| VSS | W31 |
| VSS | W32 |
| VSS | W33 |
| VSS | W38 |
| VSS | Y17 |
| VSS | Y18 |
| VSS | Y20 |
| VSS | Y21 |
| VSS | Y26 |
| VSS | Y28 |
| VSS | Y29 |
| VSS | AA4 |
| VSS | AA17 |
| VSS | AA18 |
| VSS | AA20 |
| VSS | AA21 |
| VSS | AA25 |
| VSS | AA29 |
| VSS | AA42 |
| VSS | AB4 |
| VSS | AB5 |
| VSS | AB7 |
| VSS | AB8 |
| VSS | AB10 |
| VSS | AB11 |
| VSS | AB14 |
| VSS | AB15 |
| VSS | AB31 |
| VSS | AB32 |
| VSS | AB38 |
| VSS | AC1 |
| VSS | AC18 |
| VSS | AC20 |
| VSS | AC21 |
| VSS | AC25 |
| VSS | AC29 |

### Table 11-1. BGA PCH-H Ballout (Sheet 18 of 20)

| Ball Name | Ball # |
|---|---|
| VSS | AC45 |
| VSS | AD4 |
| VSS | AD8 |
| VSS | AD11 |
| VSS | AD14 |
| VSS | AD32 |
| VSS | AD33 |
| VSS | AD36 |
| VSS | AE4 |
| VSS | AE18 |
| VSS | AE20 |
| VSS | AE21 |
| VSS | AE25 |
| VSS | AE28 |
| VSS | AE29 |
| VSS | AE42 |
| VSS | AF18 |
| VSS | AF20 |
| VSS | AF21 |
| VSS | AF23 |
| VSS | AF25 |
| VSS | AF26 |
| VSS | AF28 |
| VSS | AF29 |
| VSS | AG4 |
| VSS | AG11 |
| VSS | AG13 |
| VSS | AG31 |
| VSS | AG32 |
| VSS | AG33 |
| VSS | AG38 |
| VSS | AH1 |
| VSS | AH17 |
| VSS | AH18 |
| VSS | AH20 |
| VSS | AH21 |
| VSS | AH23 |
| VSS | AH25 |
| VSS | AH26 |
| VSS | AH28 |
| VSS | AH29 |
| VSS | AH45 |
| VSS | AJ10 |
| VSS | AJ14 |
| VSS | AJ15 |
| VSS | AJ17 |
| VSS | AJ18 |

**Table 11-1. BGA PCH-H Ballout (Sheet 19 of 20)**

| Ball Name | Ball # |
|---|---|
| VSS | AJ26 |
| VSS | AJ28 |
| VSS | AJ29 |
| VSS | AJ31 |
| VSS | AJ32 |
| VSS | AJ36 |
| VSS | AK4 |
| VSS | AK42 |
| VSS | AL4 |
| VSS | AL10 |
| VSS | AL11 |
| VSS | AL13 |
| VSS | AL17 |
| VSS | AL19 |
| VSS | AL24 |
| VSS | AL29 |
| VSS | AL32 |
| VSS | AL33 |
| VSS | AL38 |
| VSS | AM15 |
| VSS | AM17 |
| VSS | AM19 |
| VSS | AM22 |
| VSS | AM24 |
| VSS | AM27 |
| VSS | AM29 |
| VSS | AM45 |
| VSS | AN4 |
| VSS | AN7 |
| VSS | AN8 |
| VSS | AN10 |
| VSS | AN11 |
| VSS | AN22 |
| VSS | AN27 |
| VSS | AN31 |
| VSS | AN39 |
| VSS | AP4 |
| VSS | AP11 |
| VSS | AR5 |
| VSS | AR7 |
| VSS | AR9 |
| VSS | AR33 |
| VSS | AR34 |
| VSS | AR42 |
| VSS | AT9 |
| VSS | AT10 |
| VSS | AT15 |

**Table 11-1. BGA PCH-H Ballout (Sheet 20 of 20)**

| Ball Name | Ball # |
|---|---|
| VSS | AT36 |
| VSS | AU1 |
| VSS | AU7 |
| VSS | AU35 |
| VSS | AU36 |
| VSS | AU39 |
| VSS | AU45 |
| VSS | AV6 |
| VSS | AV17 |
| VSS | AV24 |
| VSS | AV27 |
| VSS | AV31 |
| VSS | AV33 |
| VSS | AW9 |
| VSS | AW13 |
| VSS | AW19 |
| VSS | AW29 |
| VSS | AW37 |
| VSS | AY38 |
| VSS | AY45 |
| VSS | BA1 |
| VSS | BB1 |
| VSS | BB11 |
| VSS | BB16 |
| VSS | BB21 |
| VSS | BB25 |
| VSS | BB30 |
| VSS | BB34 |
| VSS | BC1 |
| VSS | BC2 |
| VSS | BD2 |
| VSS | BD43 |
| VSS | BD44 |
| VSS | BD45 |
| VSS | BE9 |
| VSS | BE14 |
| VSS | BE18 |
| VSS | BE23 |
| VSS | BE28 |
| VSS | BE32 |
| VSS | BE37 |
| VSS | BE40 |
| VSS | BE44 |
| WAKE# | BC13 |
| XCLK_BIASREF | E1 |
| XTAL24_IN | A6 |
| XTAL24_OUT | A5 |

§ §

# 12 8254 Timers

## 12.1 Overview

The PCH contains two counters that have fixed uses. All registers and functions associated with the 8254 timers are in the core well. The 8254 unit is clocked by a 14.318-MHz clock derived from 24-MHz xtal clock.

**Counter 0, System Timer**

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

**Counter 2, Speaker Tone**

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

### 12.1.1 Timer Programming

The counter/timers are programmed in the following fashion:

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

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

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

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

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

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

Table 12-1 lists the six operating modes for the interval counters.

**Table 12-1. Counter Operating Modes**

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

## 12.1.2 Reading from the Interval Timer

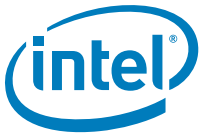
It is often desirable to read the value of a counter without disturbing the count in progress. There are three methods for reading the counters—a simple read operation, counter Latch command, and the Read-Back command. Each is explained below.

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

### 12.1.2.1 Simple Read

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

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



#### 12.1.2.2 Counter Latch Command

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

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

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

#### 12.1.2.3 Read Back Command

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

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

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

Both count and status of the selected counters may be latched simultaneously. This is functionally the same as issuing two consecutive, separate Read Back commands. If multiple counts and/or statuses 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.

§ §

# 13 Integrated High Definition Audio

## 13.1 Acronyms

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

## 13.2 References

None.

## 13.3 Overview

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

## 13.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **High Definition Audio Signals** | | |
| **HDA_RST#** | O | **HD Audio Reset**: Master H/W reset to internal/external codecs. |
| **HDA_SYNC** | O | **HD Audio Sync:** 48-KHz fixed rate frame sync to the codecs. Also used to encode the stream number. |
| **HDA_BCLK** | O | **HD Audio Bit Clock:** Up to 24-MHz serial data clock generated by the Intel HD Audio controller. |
| **HDA_SDO** | O | **HD Audio Serial Data Out:** Serial TDM data output to the codecs. The serial output is double-pumped for a bit rate of up to 48 Mb/s. |

| Name | Type | Description |
|---|---|---|
| **HDA_SDI0** | I | **HD Audio Serial Data In 0:** Serial TDM data input from the two codec(s). The serial input is single-pumped for a bit rate of up to 24 Mb/s. These signals contain integrated Pull-down resistors, which are enabled while the primary well is powered. |
| **HDA_SDI1** | I | **HD Audio Serial Data In 1:** Serial TDM data input from the two codec(s). The serial input is single-pumped for a bit rate of up to 24 Mb/s. These signals contain integrated Pull-down resistors, which are enabled while the primary well is powered. |
| **Intel Display Audio Interface** | | |
| **DISPA_BCLK** | O | **Display Audio Bit Clock:** Serial data clock generated by the Intel HD Audio controller. PCH supports data rate of up to 96 Mb/s. |
| **DISPA_SDO** | O | **Display Audio Serial Data Out:** Serial TDM data output to the codec. PCH supports data rate of up to 96 Mb/s. |
| **DISPA_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** | | |
| **I2S0_SCLK**/ GPP_D8**/** GPP_D8 | I/O | **$I^2S$/PCM serial bit clock 0:** Clock used to control the timing of a transfer. Can be generated internally (Master mode) or taken from an external source (Slave mode). |
| **I2S0_SFRM/** GPP_D5/ GPP_D5 | I/O | **$I^2S$/PCM serial frame indicator 0**: This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Master mode) or taken from an external source (Slave mode). |
| **I2S0_TXD** // GPP_D6/ GPP_D6 | O | **$I^2S$/PCM transmit data (serial data out)0**: This signal transmits serialized data. The sample length is a function of the selected serial data sample size. |
| **I2S0_RXD/** GPP_D7/ GPP_D7 | I | **$I^2S$/PCM receive data (serial data in)0:** This signal receives serialized data. The sample length is a function of the selected serial data sample size. |
| **DMIC Interface** | | |
| **DMIC_CLK0/** GPP_D19 | O | **Digital Mic Clock:** Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz. |
| **DMIC_CLK1**/ GPP_D17 | O | **Digital Mic Clock:** Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz. |
| **DMIC_DATA0** /GPP_D20 | I | **Digital Mic Data:** Serial data input from the digital mic. |
| **DMIC_DATA1** /GPP_D18 | I | **Digital Mic Data:** Serial data input from the digital mic. |

# 13.5 Integrated Pull-Ups and Pull-Downs

### Table 13-1. Integrated Pull-Ups and Pull-Downs

| Signal | Resistor Type | Value (Ω) | Notes |
|---|---|---|---|
| HDA_SYNC | Pull-down | 14K–26K | |
| HDA_SDO | Pull-down | 14K–26K | |
| HDA_SDI[1:0] | Pull-down | 14K–26K | |
| DISPA_SDO | Pull-down | 14K–26K | |
| DISPA_SDI | Pull-down | 14K–26K | |
| SSP0_SFRM | Pull-down | 14K–26K | |
| SSP0_RXD | Pull-down | 14K–26K | |
| DMIC_DATA[1:0] | Pull-down | 14K–26K | |

## 13.6 I/O Signal Planes and States

**Table 13-2. I/O Signal Planes and States**

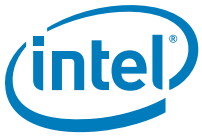
| Signal Name | Power Plane | During Reset | Immediately After Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| **High Definition Audio Interface** | | | | | |
| HDA_RST# | Primary | Driven Low | Driven Low | Driven Low | OFF |
| HDA_SYNC | Primary | Internal Pull-down | Driven Low | Internal Pull-down | OFF |
| HDA_BLK | Primary | Driven Low | Driven Low | Driven Low | OFF |
| HDA_SDO | Primary | Internal Pull-down | Driven Low | Internal Pull-down | OFF |
| HDA_SDI[1:0] | Primary | Internal Pull-down | Internal Pull-down | Internal Pull-down | OFF |
| **Display Audio Interface** | | | | | |
| DISPA_BCLK | Primary | Driven Low | Driven Low | OFF | OFF |
| DISPA_SDO | Primary | Internal Pull-down | Internal Pull-down | Internal Pull-down | OFF |
| DISPA_SDI | Primary | Internal Pull-down | Internal Pull-down | OFF | OFF |
| **$I^2S$/PCM Interface** | | | | | |
| I2S0_SCLK | Primary | Internal Pull-down | Internal Pull-down | Internal Pull-down | OFF |
| I2S0_SFRM | Primary | Internal Pull-down | Internal Pull-down | Internal Pull-down | OFF |
| I2S0_TXD | Primary | Internal Pull-down | Driven Low | Internal Pull-down | OFF |
| I2S0_RXD | Primary | Internal Pull-down | Internal Pull-down | Internal Pull-down | OFF |
| **DMIC Interface** | | | | | |
| DMIC_CLK[1:0] | Primary | Driven Low | Driven Low | Driven Low | OFF |
| DMIC_DATA[1:0] | Primary | Internal Pull-down | Internal Pull-down | Internal Pull-down | OFF |

## 13.7 Features

The Integrated High Definition Audio subsystem features are listed below.

### 13.7.1 High Definition Audio Controller Capabilities

- PCI/PCI Express controller
- 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, 9 output)
  - 16 channels per stream
  - 32 bits/sample
  - 192 KHz sample rate
- Supports memory-based command/response transport
- Supports optional Immediate Command/Response mechanism



- Supports output and input stream synchronization
- Supports global time synchronization
- Supports MSI interrupt delivery
- Support for ACPI D3 and D0 Device States
- Supports Function Level Reset (FLR)
  - Only if exposed as PCI Express device
- Supports Intel Power Optimizer Power Management
  - Support 1 ms of buffering with all DMA running with maximum bandwidth
  - Support 10 ms of buffering with 1 output DMA and 1 input DMA running at 2 channels, 96 KHz, 16-bit audio

### 13.7.2 Audio DSP Capabilities

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

### 13.7.3 High Definition Audio Link Capabilities

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

### 13.7.4 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

### 13.7.5 DSP I/O Peripherals Capabilities

- Two digital microphone ports to support up to four digital microphone modules
- 1 bi-directional $I^2S$ / PCM ports to support 1 $I^2S$ connection

§ §

# 14 Controller Link

## 14.1 Overview

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

## 14.2 Signal Description

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

## 14.3 Integrated Pull-Ups and Pull-Downs

| Signal | Resistor Type | Value (Ohm) | Notes |
|---|---|---|---|
| **CL_DATA** | Pull-up<br>Pull-down | 31.25<br>100 | See Section 14.4 |
| **CL_CLK** | Pull-up<br>Pull-down | 31.25<br>100 | See Section 14.4 |

## 14.4 I/O Signal Planes and States

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

***Notes:***
1. The Controller Link clock and data buffers use internal Pull-up or Pull-down resistors to drive a logical 1 or 0.
2. The terminated state is when the I/O buffer Pull-down is enabled.

## 14.5 Functional Description

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

§ §

# 15 Processor Sideband Signals

## 15.1 Acronyms

| Acronyms | Description |
|---|---|
| PECI | Platform Environmental Control Interface |

## 15.2 Overview

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

## 15.3 Signal Description

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

## 15.4 Integrated Pull-Ups and Pull-Downs

None

## 15.5 I/O Signal Planes and States

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| **PROCPWRGD** | Primary | Driven Low | Driven High | Off | Off |
| **THERMTRIP#** | Primary | Undriven | Undriven | Off | Off |
| **PM_SYNC** | Primary | Driven Low | Driven Low | Off | Off |
| **PM_DOWN** | Primary | Undriven | Undriven | Off | Off |
| **PLTRST_PROC#** | Primary | Driven Low | Driven High | Off | Off |

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| **PECI** | Primary | Undriven | Undriven | Off | Off |
| **CPU_GP[3:0]** | Primary | Undriven | Undriven | Undriven | Off |

## 15.6 Functional Description

PROCPWRGD out to the processor indicates that the primary power is ramped up and stable.

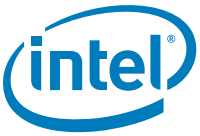
If THERMTRIP# goes active, the processor is indicating an overheat condition, and the PCH will immediately transition to an S5 state. CPU_GP can be used from external sensors for the thermal management.

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.

PLTRST_PROC# is the platform reset to the processor.

§ §



# 16 Digital Display Signals

## 16.1 Acronyms

| Acronyms | Description |
|---|---|
| eDP* | embedded Display Port* |

## 16.2 References

None

## 16.3 Signal Description

Display is divided between processor and PCH. The processor houses memory interface, display planes, pipes, and digital display interfaces/ports while the PCH has transcoder and analog display interface or port.

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

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

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

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

**Table 16-1. Digital Display Signals (Sheet 1 of 2)**

| Name | Type | Description |
|---|---|---|
| **DDPB_HPD0** / GPP_I0 | I | **Display Port B:** HPD Hot-Plug Detect |
| **DDPC_HPD1** / GPP_I1 | I | **Display Port C:** HPD Hot-Plug Detect |
| **DDPD_HPD2** / GPP_I2 | I | **Display Port D:** HPD Hot-Plug Detect<br>or eDP[1] Hot Plug Detect |
| **DDPE_HPD3** / GPP_I3 | I | **Display Port E:** HPD Hot-Plug Detect |
| DDPB_CTRLCLK / GPP_I5 | I/O | **Display Port B:** Control Clock. |
| **DDPB_CTRLDATA** / GPP_I6 | I/O | **Display Port B:** Control Data. |
| **DDPC_CTRLCLK** / GPP_I7 | I/O | **Display Port C:** Control Clock |
| **DDPC_CTRLDATA** / GPP_I8 | I/O | **Display Port C:** Control Data |

**Table 16-1. Digital Display Signals (Sheet 2 of 2)**

| Name | Type | Description |
|---|---|---|
| **DDPD_CTRLCLK** / GPP_I9 | I/O | **Display Port D:** Control Clock |
| **DDPD_CTRLDATA** / GPP_I10 | I/O | **Display Port D:** Control Data |

## 16.4 Embedded DisplayPort* (eDP*) Backlight Control Signals

| | | |
|---|---|---|
| **eDP_VDDEN** / GPP_F19 | O | **eDP Panel power Enable:** Panel power control enable.<br>This signal is used to control the VDC source of the panel logic. |
| **eDP_BKLTEN** / GPP_F20 | O | **eDP Backlight Enable:** Panel backlight enable control for eDP<br>This signal is used to gate power into the backlight circuitry. |
| **eDP_BKLTCTL** / GPP_F21 | O | **eDP Panel Backlight Brightness control:** Panel brightness control for eDP.<br>This signal is used as the PWM Clock input signal |
| **EDP_HPD** / GPP_I4 | I | **eDP:** Hot-Plug Detect |
| ***Note:*** eDP_VDDEN, eDP_BKLTEN, eDP_BKLTCTL can be left as no connect if eDP* is not used. | | |

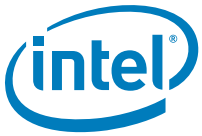
## 16.5 Integrated Pull-Ups and Pull-Downs

| Signal | Resistor Type | Value | Notes |
|---|---|---|---|
| DDPB_CTRLDATA | Pull-down | 15K-40K | see note below |
| DDPC_CTRLDATA | Pull-down | 15K-40K | see note below |
| DDPD_CTRLDATA | Pull-down | 15K-40K | see note below |

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

## 16.6 I/O Signal Planes and States

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



| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| **eDP_VDDEN** | Primary | Driven Low | Driven Low | Driven Low | Off |
| **eDP_BKLTEN** | Primary | Driven Low | Driven Low | Driven Low | Off |
| **eDP_BKLTCTL** | Primary | Driven Low | Driven Low | Driven Low | Off |
| **EDP_HPD** | Primary | Undriven | Undriven | Undriven | Off |

§ §

# 17 Enhanced Serial Peripheral Interface (eSPI)

## 17.1 Acronyms

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

## 17.2 References

None.

## 17.3 Overview

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

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

***Note:*** For the PCH Server SKU, a second chip select is available to support up to 2 eSPI slave devices.

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

## 17.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **ESPI_CLK/** CLKOUT_LPC0/ GPP_A9 | O | **eSPI Clock:** eSPI clock output from the PCH to slave device. |
| **ESPI_IO0/** LAD0/GPP_A1 | I/O | **eSPI Data Signal 0:** Bi-directional pin used to transfer data between the PCH and eSPI slave device. |
| **ESPI_IO1/** LAD1/GPP_A2 | I/O | **eSPI Data Signal 1:** Bi-directional pin used to transfer data between the PCH and eSPI slave device |
| **ESPI_IO2/** LAD2/GPP_A3 | I/O | **eSPI Data Signal 2:** Bi-directional pin used to transfer data between the PCH and eSPI slave device |
| **ESPI_IO3/** LAD3/GPP_A4 | I/O | **eSPI Data Signal 3:** Bi-directional pin used to transfer data between the PCH and eSPI slave device |

| Name | Type | Description |
|---|---|---|
| **ESPI_CS#/** LFRAME#/ GPP_A5 | O | **eSPI Chip Select 0:** Driving CS# signal low to select eSPI slave for the transaction. |
| **ESPI_CS1# (Server Only) /** SERIRQ/ GPP_A6 | O | **eSPI Chip Select 1 (Server Only):** Driving CS# signal low to select eSPI slave for the transaction. |
| **ESPI_ALERT0# (Server Only)** / PIRQA#/ GPP_A7 | I | **eSPI Alert 0 (Server Only):** Alert signal from eSPI slave to the PCH.<br>***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. |
| **ESPI_ALERT1# (Server Only)** / RCIN#/ GPP_A0 | I | **eSPI Alert 1 (Server Only):** Alert signal from eSPI slave to the PCH.<br>***Note:*** If only a single Slave is connected, the eSPI Compatibility Spec requires that the Slave must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin. |
| **ESPI_RST#/** SUS_STAT#/ GPP_A14 | O | **eSPI Reset:** Reset signal from the PCH to eSPI slave. |

## 17.5 Integrated Pull-Ups and Pull-Downs

| Signal | Resistor Type | Value | Notes |
|---|---|---|---|
| **ESPI_CLK** | Pull-down | 9K - 50K | |
| **ESPI_IO[3:0]** | Pull-up | 15K - 40K | |
| **ESPI_ CS [1:0]#** | Pull-up | 15K - 40K | |
| **ESPI_ALERT [1:0]#** | Pull-up | 15K - 40K | |

## 17.6 I/O Signal Planes and States

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

## 17.7 Functional Description

### 17.7.1 Features

The PCH eSPI controller supports the following features:

- Master mode only, allowing one slave device to be connected to the PCH
  Note: For PCH Server SKU, a second chip select signal is available allowing the second slave to be connected.

- Support for 20 MHz, 24 MHz, 30 MHz, 48 MHz, and 60 MHz (configured by soft straps)
- 1.8V support only
- Up to quad mode support
- In-band messages for communication between the PCH and 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

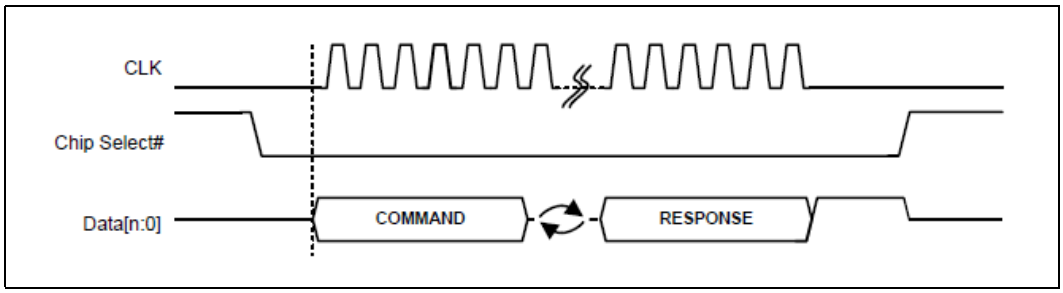
***Note:*** For client platform, the PCH eSPI controller does not support a discrete ALERT# pin (as described in the eSPI specification) since the PCH supports only a Single Master - Single Slave configuration. Only ALERT# signaling (over ESPI_IO1) is supported.

For Server, the PCH eSPI support two ALERT# pins to support alerts from two supported slave devices to the PCH. However, note that if only a single slave is connected, the eSPI Compatibility Spec requires that the Slave must operate with in-band Alert# signaling in order to free up the GPIO pin required for the discrete Alert# pin.

## 17.7.2 Protocols

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

**Figure 17-1. Basic eSPI Protocol**

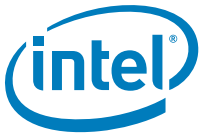



An eSPI transaction consists of a Command phase driven by the master, a turn-around phase (TAR), and a Response phase driven by the slave.

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

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



### 17.7.3 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.

### 17.7.4 In-Band Link Reset

In case the eSPI link may end up in an undefined state (for example when a CRC error is received from the 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.

### 17.7.5 Slave Discovery

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

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

### 17.7.6 Channels and Supported Transactions

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

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

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

Table 17-1 summarizes the eSPI channels and supported transactions.

**Table 17-1. eSPI Channels and Supported Transactions**

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

#### 17.7.6.1 Peripheral Channel (Channel 0) Overview

The Peripheral channel performs the following Functions:

- Target for PCI Device D31:F0: The eSPI controller duplicates the legacy LPC PCI Configuration space registers. These registers are mostly accessed via the BIOS, though some are accessed via the OS as well.
- Tunnel all Host to 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.

### 17.7.6.2 Virtual Wire Channel (Channel 1) Overview

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

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

#### 17.7.6.2.1 eSPI Virtual Wires (VW)

Table 17-2 summarizes the PCH virtual wires in eSPI mode.

**Table 17-2. eSPI Virtual Wires (VW)**

| Virtual Wire | PCH Pin Direction | Reset Control | Pin Retained in PCH (For Use by Other Components) |
|---|---|---|---|
| SUS_STAT# | Output | ESPI_RESET# | No |
| SUS_PWRDN_ACK | Output | ESPI_RESET# | No |
| PLTRST# | Output | ESPI_RESET# | Yes |
| PME# | Input | ESPI_RESET# | No |
| WAKE# | Input | ESPI_RESET# | No |
| SMI# | Input | PLTRST# | N/A |
| SCI# | Input | PLTRST# | N/A |
| RCIN# | Input | PLTRST# | No |
| SLP_A# | Output | ESPI_RESET# | Yes |
| SLP_S3#/SLP_S4#/ SLP_S5#/SLP_LAN#/ SLP_WLAN# | Output | DSW_PWROK | Yes |

#### 17.7.6.2.2 Interrupt Events

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

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

The 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.

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

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

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

#### 17.7.6.3.1 PCH Temperature Data Over eSPI OOB Channel

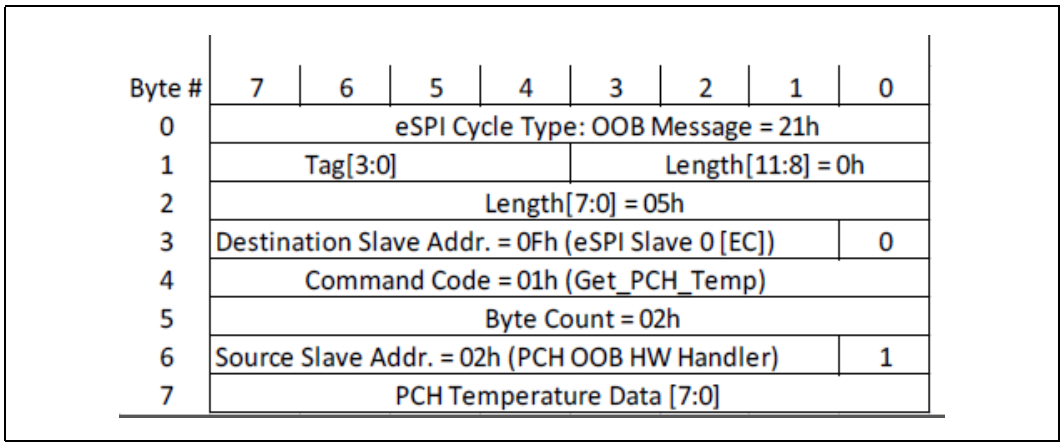
eSPI controller supports the transmitting of PCH thermal data to the eSPI slave. The thermal data consists of 1 byte of PCH temperature data that is transmitted periodically (~1 ms) from the thermal sensor unit.

The packet formats for the temperature request from the eSPI slave and the PCH response back are shown in Figure 17-2 and Figure 17-3.

**Figure 17-2. eSPI Slave Request to PCH for PCH Temperature**

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

**Figure 17-3. PCH Response to eSPI Slave with PCH Temperature**



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

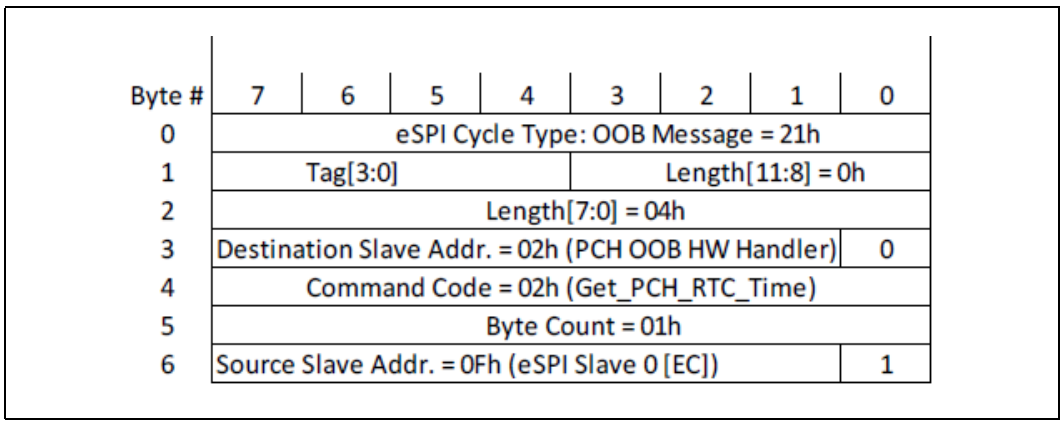
### 17.7.6.3.2 PCH RTC Time/Date to EC Over eSPI OOB Channel

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

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

The packet formats for the RTC time/date request from the eSPI slave and the PCH response back to the device are shown in Figure 17-4 and Figure 17-5.

**Figure 17-4. eSPI Slave Request to PCH for PCH RTC Time**



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

**Figure 17-5. PCH Response to eSPI Slave with RTC Time**

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

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

### 17.7.6.4 Flash Access Channel (Channel 3) Overview

The eSPI flash access channel supports 2 different kinds of flash access:

1. 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.
2. Slave Attached Flash (SAF) (**SERVER ONLY**): In this configuration, the flash device is attached to the BMC. All PCH flash accesses are routed over eSPI to the BMC which performs the requested flash operation and returns a completion back to the PCH.

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.

#### 17.7.6.4.1 Master Attached Flash Channel Controller (MAFCC) Flash Operations and Addressing

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

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

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

#### 17.7.6.4.2 Slave Attached Flash Channel Controller (SAFCC) Flash Operation and Addressing (SERVER ONLY)

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

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

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

§ §

# 18 General Purpose Input and Output (GPIO)

## 18.1 Acronyms

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

## 18.2 References

None

## 18.3 Overview

The PCH General Purpose Input/Output (GPIO) signals are grouped into multiple groups (such as GPP_A, GPP_B, and so on) and are powered by either the PCH Primary well or Deep Sleep well. Each of these pin groups has a dedicated power pin that can be set to either 1.8V or 3.3V. All pins within the same group (including the native functionality that is multiplexed with the GPIO) operate at the same voltage determined by the power supplied to the power pins.

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

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

**Table 18-1. GPIO Group Summary**

| GPIO Group | Power Pins | Voltage |
|---|---|---|
| Primary Well Group A (GPP_A) | VCCPGPPA | 1.8V or 3.3V |
| Primary Well Group B (GPP_B)<br>Primary Well Group C (GPP_C)<br>Primary Well Group H (GPP_H) | VCCPGPPBCH | 1.8V or 3.3V |
| Primary Well Group D (GPP_D) | VCCPGPPD | 1.8V or 3.3V |
| Primary Well Group E (GPP_E)<br>Primary Well Group F (GPP_F) | VCCPGPPEF | 1.8V or 3.3V |
| Primary Well Group G (GPP_G) | VCCPGPPG | 1.8V or 3.3V |
| Primary Well Group I (GPP_I) | VCCPRIM_3P3 | 3.3V |
| Deep Sleep Well Group (GPD) | VCCDSW_3P3 | 3.3V |

## 18.4 Signal Description

Table 18-2 summarizes the GPIO implementation in the PCH.

**Table 18-2. General Purpose I/O Signals (Sheet 1 of 12)**

| Name | Internal Pull-up/ Pull-down (Note 1) | De-Glitch | | Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 2) | Default | NMI or SMI Capable | Note |
|---|---|---|---|---|---|---|---|
| | | Input | Output | | | | |
| **Group A GPIO - Primary Power Well (1.8V or 3.3V)** | | | | | | | |
| **GPP_A0** | None | No | Yes (Note 4) | LPC Mode: RCIN#(1st) eSPI Mode: ESPI_ALERT1# (3rd) (Server/WS Only) | RCIN# (LPC mode) GPI (eSPI mode) ESPI_ALERT1#(eSPI mode in Server SKU Only) | None | See Note 8 |
| **GPP_A1** | None | No | Yes (Note 4) | LPC mode: LAD0 (1st) eSPI mode: ESPI_IO0 (3rd) | LAD0 (LPC mode) ESPI_IO0 (eSPI mode) | None | See Note 8 |
| **GPP_A2** | None | No | Yes (Note 4) | LPC mode: LAD1 (1st) eSPI mode: ESPI_IO1 (3rd) | LAD1 (LPC mode) ESPI_IO1 (eSPI mode) | None | See Note 8 |
| **GPP_A3** | None | No | Yes (Note 4) | LPC mode: LAD2 (1st) eSPI mode: ESPI_IO2 (3rd) | LAD2 (LPC mode) ESPI_IO2 (eSPI mode) | None | See Note 8 |
| **GPP_A4** | None | No | Yes (Note 4) | LPC mode: LAD3 (1st) eSPI mode: ESPI_IO3 (3rd) | LAD3 (LPC mode) ESPI_IO3 (eSPI mode) | None | See Note 8 |
| **GPP_A5** | None | No | Yes (Note 5) | LPC mode: LFRAME# (1st) eSPI mode: ESPI_CS# (3rd) | LFRAME# (LPC mode) ESPI_CS# (eSPI mode) | None | See Note 8 |
| **GPP_A6** | None | No | Yes (Note 4) | LPC Mode: SERIRQ (1st) eSPI Mode: ESPI_CS1# (3rd) (Server/WS Only) | SERIRQ (LPC mode) GPI (eSPI mode) ESPI_CS#1 (eSPI mode in Server SKU Only) | None | See Note 8 |
| **GPP_A7** | None | No | Yes (Note 4) | LPC Mode: PIRQA# (1st) eSPI Mode: ESPI_ALERT0# (3rd) (Server/WS Only) | PIRQA# (LPC mode) GPI (eSPI mode) ESPI_ALERT0# (eSPI mode in Server SKU Only) | None | See Note 8 |
| **GPP_A8** | None | No | Yes (Note 4) | LPC Mode: CLKRUN# eSPI Mode: None | CLKRUN# (LPC mode) GPI (eSPI mode) CLKRUN# (eSPI mode in Server SKU Only) | None | See Note 8 |

## Table 18-2. General Purpose I/O Signals (Sheet 2 of 12)

| Name | Internal Pull-up/ Pull-down (Note 1) | De-Glitch | | Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 2) | Default | NMI or SMI Capable | Note |
|---|---|---|---|---|---|---|---|
| | | Input | Output | | | | |
| **GPP_A9** | None | No | Yes (Note 5) | LPC mode: CLKOUT_LPC0 (1st)<br>eSPI mode: ESPI_CLK (3rd) | CLKOUT_LPC0 (LPC Mode)<br>ESPI_CLK (eSPI mode) | None | See Note 8 |
| **GPP_A10** | None | No | Yes (Note 4) | LPC Mode: CLKOUT_LPC1<br>eSPI Mode: None | CLKOUT_LPC1 (LPC mode)<br>GPI (eSPI mode)<br>CLKOUT_LPC1(eSPI mode in Server SKU Only) | None | See Note 8 |
| **GPP_A11** | None | No | Yes (Note 4) | LPC mode: PME#<br>eSPI mode: None | PME# (LPC mode)<br>GPI (eSPI mode) | None | See Note 8 |
| **GPP_A12** | None | No | Yes (Note 4) | BM_BUSY# (1st) / ISH_GP6 (2nd) / SX_EXIT_HOLDOFF# (3rd) | GPI | None | |
| **GPP_A13** | None | No | Yes (Note 5) | LPC mode: SUSWARN# / SUSPWRDNACK<br>eSPI mode: None | SUSWANRN#/ SUSPWRDNACK (LPC mode)<br>GPI (eSPI mode) | None | See Note 8 |
| **GPP_A14** | None | No | Yes (Note 5) | LPC mode: SUS_STAT# (1st)<br>eSPI mode: ESPI_RESET# (3rd) | SUS_STAT# (LPC mode)<br>ESPI_RESET# (eSPI mode) | None | See Note 8 |
| **GPP_A15** | None | No | Yes (Note 4) | LPC mode: SUS_ACK#<br>eSPI mode: None | SUS_ACK# (LPC mode)<br>GPI (eSPI mode) | None | See Note 8 |
| **GPP_A16** | None | No | Yes (Note 4) | CLKOUT_48 (1st) (Server SKU Only) | Native or GPI | None | Default depends on soft strap |
| **GPP_A17** | None | No | Yes (Note 4) | ISH_GP7 | GPI | None | |
| **GPP_A18** | None | No | Yes (Note 4) | ISH_GP0 | GPI | None | |
| **GPP_A19** | None | No | Yes (Note 4) | ISH_GP1 | GPI | None | |
| **GPP_A20** | None | No | Yes (Note 4) | ISH_GP2 | GPI | None | |
| **GPP_A21** | None | No | Yes (Note 4) | ISH_GP3 | GPI | None | |
| **GPP_A22** | None | No | Yes (Note 4) | ISH_GP4 | GPI | None | |
| **GPP_A23** | None | No | Yes (Note 4) | ISH_GP5 | GPI | None | |
| **Group B GPIO - Primary Power Well (1.8V or 3.3V)** | | | | | | | |
| **GPP_B0** | None | No | Yes (Note 5) | None | GPO | None | As GPO, the signal defaults to '0' |

## Table 18-2. General Purpose I/O Signals (Sheet 3 of 12)

| Name | Internal Pull-up/ Pull-down (Note 1) | De-Glitch | | Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 2) | Default | NMI or SMI Capable | Note |
|---|---|---|---|---|---|---|---|
| | | Input | Output | | | | |
| **GPP_B1** | None | No | Yes (Note 5) | None | GPO | None | As GPO, the signal defaults to '0' |
| **GPP_B2** | None | No | Yes (Note 4) | VRALERT# | GPI | None | |
| **GPP_B3** | None | No | Yes (Note 4) | CPU_GP2 | GPI | None | |
| **GPP_B4** | None | No | Yes (Note 4) | CPU_GP3 | GPI | None | |
| **GPP_B5** | None | No | Yes (Note 4) | SRCCLKREQ0# | GPI | None | |
| **GPP_B6** | None | No | Yes (Note 4) | SRCCLKREQ1# | GPI | None | |
| **GPP_B7** | None | No | Yes (Note 4) | SRCCLKREQ2# | GPI | None | |
| **GPP_B8** | None | No | Yes (Note 4) | SRCCLKREQ3# | GPI | None | |
| **GPP_B9** | None | No | Yes (Note 4) | SRCCLKREQ4# | GPI | None | |
| **GPP_B10** | None | No | Yes (Note 4) | SRCCLKREQ5# | GPI | None | |
| **GPP_B11** | None | No | Yes (Note 5) | None | GPO | None | As GPO, the signal defaults to '0' |
| **GPP_B12** | None | No | Yes (Note 6) | SLP_S0# | SLP_S0# | None | |
| **GPP_B13** | None | No | Yes (Note 5) | PLTRST# | PLTRST# | None | |
| **GPP_B14** | 20K PD (Note 3) | No | No | SPKR | GPO | NMI SMI | •Also used as a strap.<br>•The pull-down resistor is disabled after PCH_PWROK de-asserts<br>•As GPO, the signal defaults to '0' |
| **GPP_B15** | None | No | Yes (Note 4) | GSPI0_CS# | GPI | None | |
| **GPP_B16** | None | No | Yes (Note 4) | GSPI0_CLK | GPI | None | |
| **GPP_B17** | None | No | Yes (Note 4) | GSPI0_MISO | GPI | None | |
| **GPP_B18** | 20K PD (Note 3) | No | No | GSPI0_MOSI | GPO | None | •Also used as a strap.<br>•The pull-down resistor is disabled after PCH_PWROK de-asserts<br>•As GPO, the signal defaults to '0' |
| **GPP_B19** | None | No | Yes (Note 4) | GSPI1_CS# | GPI | None | |
| **GPP_B20** | None | No | Yes (Note 4) | GSPI1_CLK | GPI | NMI SMI | |
| **GPP_B21** | None | No | Yes (Note 4) | GSPI1_MISO | GPI | None | |

## Table 18-2. General Purpose I/O Signals (Sheet 4 of 12)

| Name | Internal Pull-up/ Pull-down (Note 1) | De-Glitch | | Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 2) | Default | NMI or SMI Capable | Note |
|---|---|---|---|---|---|---|---|
| | | Input | Output | | | | |
| **GPP_B22** | 20K PD (Note 3) | No | No | GSPI1_MOSI | GPO | None | •Also used as a strap.<br>•The pull-down resistor is disabled after PCH_PWROK de-asserts<br>•As GPO, the signal defaults to '0' |
| **GPP_B23** | 20K PD (Note 3) | Yes (Note 7) | No | SML1ALERT# (1st) / PCHHOT# (2nd) | GPO | NMI<br>SMI | •The pull-down resistor is disabled after RSMRST# de-asserts<br>•As GPO, the signal defaults to '0' |
| **Group C GPIO - Primary Power Well (1.8V or 3.3V)** | | | | | | | |
| **GPP_C0** | None | Yes (Note 7) | Yes (Note 4) | SMBCLK | SMBCLK | None | |
| **GPP_C1** | None | Yes (Note 7) | Yes (Note 4) | SMBDATA | SMBDATA | None | |
| **GPP_C2** | 20K PD (Note 3 | Yes (Note 7) | No | SMBALERT# | GP0 | None | •Also used as a strap.<br>•The pull-down resistor is disabled after RSMRST# de-asserts<br>•As GPO, the signal defaults to '0' |
| **GPP_C3** | None | Yes (Note 7) | Yes (Note 4) | SML0CLK | SML0CLK | None | |
| **GPP_C4** | None | Yes (Note 7) | Yes (Note 4) | SML0DATA | SML0DATA | None | |
| **GPP_C5** | 20K PD (Note 3) | Yes (Note 7) | No | SML0ALERT# | GPO | None | •Also used as a strap.<br>•The pull-down resistor is disabled after RSMRST# de-asserts<br>•As GPO, the signal defaults to '0' |
| **GPP_C6** | None | Yes (Note 7) | Yes (Note 4) | SML1CLK | GPI | None | |
| **GPP_C7** | None | Yes (Note 7) | Yes (Note 4) | SML1DATA | GPI | None | |
| **GPP_C8** | None | No | Yes (Note 4) | UART0_RXD | GPI | None | |
| **GPP_C9** | None | No | Yes (Note 4) | UART0_TXD | GPI | None | |
| **GPP_C10** | None | No | Yes (Note 4) | UART0_RTS# | GPI | None | |
| **GPP_C11** | None | No | Yes (Note 4) | UART0_CTS# | GPI | None | |
| **GPP_C12** | None | No | Yes (Note 4) | UART1_RXD (1st)/ ISH_UART1_RXD (2nd) | GPI | None | |
| **GPP_C13** | None | No | Yes (Note 4) | UART1_TXD (1st) / ISH_UART1_TXD (2nd) | GPI | None | |

## Table 18-2. General Purpose I/O Signals (Sheet 5 of 12)

| Name | Internal Pull-up/ Pull-down (Note 1) | De-Glitch | | Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 2) | Default | NMI or SMI Capable | Note |
|---|---|---|---|---|---|---|---|
| | | Input | Output | | | | |
| **GPP_C14** | None | No | Yes (Note 4) | UART1_RTS# (1st) ISH_UART1_RTS# (2nd) | GPI | None | |
| **GPP_C15** | None | No | Yes (Note 4) | UART1_CTS# (1st) ISH_UART1_CTS# (2nd) | GPI | None | |
| **GPP_C16** | None | Yes (Note 7) | Yes (Note 4) | I2C0_SDA | GPI | None | |
| **GPP_C17** | None | Yes (Note 7) | Yes (Note 4) | I2C0_SCL | GPI | None | |
| **GPP_C18** | None | Yes (Note 7) | Yes (Note 4) | I2C1_SDA | GPI | None | |
| **GPP_C19** | None | Yes (Note 7) | Yes (Note 4) | I2C1_SCL | GPI | None | |
| **GPP_C20** | None | No | Yes (Note 4) | UART2_RXD | GPI | None | |
| **GPP_C21** | None | No | Yes (Note 4) | UART2_TXD | GPI | None | |
| **GPP_C22** | None | No | Yes (Note 4) | UART2_RTS# | GPI | NMI SMI | |
| **GPP_C23** | None | No | Yes (Note 4) | UART2_CTS# | GPI | NMI SMI | |
| **Group D GPIO in Primary Power Well (1.8V or 3.3V)** | | | | | | | |
| **GPP_D0** | None | No | Yes (Note 4) | None | GPI | NMI SMI | This GPIO is blink capable |
| **GPP_D1** | None | No | Yes (Note 4) | None | GPI | NMI SMI | This GPIO is blink capable |
| **GPP_D2** | None | No | Yes (Note 4) | None | GPI | NMI SMI | This GPIO is blink capable |
| **GPP_D3** | None | No | Yes (Note 4) | None | GPI | NMI SMI | This GPIO is blink capable |
| **GPP_D4** | None | Yes (Note 7) | Yes (Note 4) | ISH_I2C2_SDA (1st) / I2C3_SDA (2nd) | GPI | NMI SMI | This GPIO is blink capable |
| **GPP_D5** | None | No | Yes (Note 4) | I2S_SFRM | GPI | None | |
| **GPP_D6** | None | No | Yes (Note 4) | I2S_TXD | GPI | None | |
| **GPP_D7** | None | No | Yes (Note 4) | I2S_RXD | GPI | None | |
| **GPP_D8** | None | No | Yes (Note 4) | I2S_SCLK | GPI | None | |
| **GPP_D9** | None | No | Yes (Note 4) | None | GPI | None | |
| **GPP_D10** | None | No | Yes (Note 4) | None | GPI | None | |
| **GPP_D11** | None | No | Yes (Note 4) | None | GPI | None | |

## Table 18-2. General Purpose I/O Signals (Sheet 6 of 12)

| Name | Internal Pull-up/ Pull-down (Note 1) | De-Glitch | | Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 2) | Default | NMI or SMI Capable | Note |
|---|---|---|---|---|---|---|---|
| | | Input | Output | | | | |
| **GPP_D12** | None | No | Yes (Note 4) | None | GPI | None | |
| **GPP_D13** | None | No | Yes (Note 4) | ISH_UART0_RXD 1st) / I2C_SDA (3rd) | GPI | None | |
| **GPP_D14** | None | No | Yes (Note 4) | ISH_UART0_TXD (1st) / I2C2_SCL (3rd) | GPI | None | |
| **GPP_D15** | None | No | Yes (Note 4) | ISH_UART0_RTS# | GPI | None | |
| **GPP_D16** | None | No | Yes (Note 4) | ISH_UART0_CTS# | GPI | None | |
| **GPP_D17** | None | No | Yes (Note 4) | DMIC_CLK1 | GPI | None | |
| **GPP_D18** | None | No | Yes (Note 4) | DMIC_DATA1 | GPI | None | |
| **GPP_D19** | None | No | Yes (Note 4) | DMIC_CLK0 | GPI | None | |
| **GPP_D20** | None | No | Yes (Note 4) | DMIC_DATA0 | GPI | None | |
| **GPP_D21** | None | No | Yes (Note 4) | None | GPI | None | |
| **GPP_D22** | None | No | Yes (Note 4) | None | GPI | None | |
| **GPP_D23** | None | Yes (Note 7) | Yes (Note 4) | ISH_I2C2_SCL (1st) / I2C3_SCL (2nd) | GPI | None | |
| **Group E GPIO - Primary Power Well (1.8V or 3.3V)** | | | | | | | |
| **GPP_E0** | None | No | Yes (Note 4) | SATAXPCIE0 (1st)/ SATAGP0 (2nd) | SATAXPCIE0 / SATAGP0 or GPI | NMI SMI | Default SATAXPCIE0 is set by a soft strap. Default is GPI before soft straps are loaded |
| **GPP_E1** | None | No | Yes (Note 4) | SATAXPCIE1 (1st)/ SATAGP1 (2nd) | SATAXPCIE1/ SATAGP1 or GPI | NMI SMI | Default SATAXPCIE1 is set by a soft strap. Default is GPI before soft straps are loaded |
| **GPP_E2** | None | No | Yes (Note 4) | SATAXPCIE2 (1st)/ SATAGP2 (2nd) | SATAXPCIE2/ SATAGP2 or GPI | NMI SMI | Default SATAXPCIE2 is set by a soft strap. Default state is GPI before soft straps are loaded |
| **GPP_E3** | None | No | Yes (Note 4) | CPU_GP0 | GPI | NMI SMI | |
| **GPP_E4** | None | No | Yes (Note 4) | SATA_DEVSLP0 | GPI | NMI SMI | |
| **GPP_E5** | None | No | Yes (Note 4) | SATA_DEVSLP1 | GPI | NMI SMI | |
| **GPP_E6** | None | No | Yes (Note 4) | SATA_DEVSLP2 | GPI | NMI SMI | |

## Table 18-2. General Purpose I/O Signals (Sheet 7 of 12)

| Name | Internal Pull-up/ Pull-down (Note 1) | De-Glitch | | Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 2) | Default | NMI or SMI Capable | Note |
|---|---|---|---|---|---|---|---|
| | | Input | Output | | | | |
| **GPP_E7** | None | No | Yes (Note 4) | CPU_GP1 | GPI | NMI SMI | |
| **GPP_E8** | None | No | Yes (Note 4) | SATA_LED# | GPI | NMI SMI | |
| **GPP_E9** | 20K PD (See note) | No | Yes (Note 4) | USB_OC0# | GPI | None | •The pull-down resistor is disabled after RSMRST# de-asserts |
| **GPP_E10** | 20K PD (Note 3) | No | Yes (Note 4) | USB_OC1# | GPI | None | •The pull-down resistor is disabled after RSMRST# de-asserts |
| **GPP_E11** | 20K PD (Note 3) | No | Yes (Note 4) | USB_OC2# | GPI | None | •The pull-down resistor is disabled after RSMRST# de-asserts |
| **GPP_E12** | 20K PD (Note 3) | No | Yes (Note 4) | USB_OC3# | GPI | None | •The pull-down resistor is disabled after RSMRST# de-asserts |
| **Group F GPIO - Primary Power Well (1.8V or 3.3V)** | | | | | | | |
| **GPP_F0** | None | No | Yes (Note 4) | SATAXPCIE3 (1st)/ SATAGP3 (2nd) | SATAXPCIE3 / SATAGP3 or GPI | None | Default SATAXPCIE3 is set by a soft strap. Default is GPI before soft straps are loaded |
| **GPP_F1** | None | No | Yes (Note 4) | SATAXPCIE4 (1st)/ SATAGP4 (2nd) | SATAXPCIE4 / SATAGP4 or GPI | None | Default SATAXPCIE4 is set by a soft strap. Default is GPI before soft straps are loaded |
| **GPP_F2** | None | No | Yes (Note 4) | SATAXPCIE5 (1st) / SATAGP5 (2nd) | SATAXPCIE5 / SATAGP5 or GPI | None | Default SATAXPCIE5 is set by a soft strap. Default is GPI before soft straps are loaded |
| **GPP_F3** | None | No | Yes (Note 4) | SATAXPCIE6 (1st) / SATAGP6 (2nd) (Server/WS Only) | SATAXPCIE6 / SATAGP6 or GPI | None | Default SATAXPCI6 is set by a soft strap. Default is GPI before soft straps are loaded |
| **GPP_F4** | None | No | Yes (Note 4) | SATAXPCIE7 (1st)/ SATAGP7 (2nd) (Server/WS Only) | SATAXPCIE7 / SATAGP7 or GPI | None | Default SATAXPCI7 is set by a soft strap. Default is GPI before soft straps are loaded |
| **GPP_F5** | None | No | Yes (Note 4) | SATA_DEVSLP3 | GPI | None | |
| **GPP_F6** | None | No | Yes (Note 4) | SATA_DEVSLP4 | GPI | None | |
| **GPP_F7** | None | No | Yes (Note 4) | SATA_DEVSLP5 | GPI | None | |
| **GPP_F8** | None | No | Yes (Note 4) | SATA_DEVSLP6 (Server/WS Only) | GPI | None | |
| **GPP_F9** | None | No | Yes (Note 4) | SATA_DEVSLP7 (Server/WS Only) | GPI | None | |
| **GPP_F10** | None | No | Yes (Note 4) | SATA_SCLOCK | GPI | None | |

## Table 18-2. General Purpose I/O Signals (Sheet 8 of 12)

| Name | Internal Pull-up/ Pull-down (Note 1) | De-Glitch | | Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 2) | Default | NMI or SMI Capable | Note |
|---|---|---|---|---|---|---|---|
| | | Input | Output | | | | |
| **GPP_F11** | None | No | Yes (Note 4) | SATA_SLOAD | GPI | None | |
| **GPP_F12** | None | No | Yes (Note 4) | SATA_SDATAOUT1 | GPI | None | |
| **GPP_F13** | None | No | Yes (Note 4) | SATA_SDATAOUT2 | GPI | None | |
| **GPP_F14** | None | No | Yes (Note 4) | None | GPI | None | |
| **GPP_F15** | None | No | Yes (Note 4) | USB_OC4# | GPI | None | |
| **GPP_F16** | None | No | Yes (Note 4) | USB_OC5# | GPI | None | |
| **GPP_F17** | None | No | Yes (Note 4) | USB_OC6# | GPI | None | |
| **GPP_F18** | None | No | Yes (Note 4) | USB_OC7# | GPI | None | |
| **GPP_F19** | None | No | Yes (Note 4) | eDP_VDDEN | GPI | None | |
| **GPP_F20** | None | No | Yes (Note 4) | eDP_BKLTEN | GPI | None | |
| **GPP_F21** | None | No | Yes (Note 4) | eDP_BKLTCTL | GPI | None | |
| **GPP_F22** | None | No | Yes (Note 4) | None | GPI | None | |
| **GPP_F23** | None | No | Yes (Note 4) | None | GPI | None | |
| **Group G GPIO - Primary Power Well (1.8V or 3.3V)** | | | | | | | |
| **GPP_G0** | None | Yes (Note 7) | Yes (Note 4) | FAN_TACH_0 (Server Only) | GPI | None | |
| **GPP_G1** | None | Yes (Note 7) | Yes (Note 4) | FAN_TACH_1 (Server Only) | GPI | None | |
| **GPP_G2** | None | Yes (Note 7) | Yes (Note 4) | FAN_TACH_2 (Server Only) | GPI | None | |
| **GPP_G3** | None | Yes (Note 7) | Yes (Note 4) | FAN_TACH_3 (Server Only) | GPI | None | |
| **GPP_G4** | None | Yes (Note 7) | Yes (Note 4) | FAN_TACH_4 (Server Only) | GPI | None | |
| **GPP_G5** | None | Yes (Note 7) | Yes (Note 4) | FAN_TACH_5 (Server Only) | GPI | None | |
| **GPP_G6** | None | Yes (Note 7) | Yes (Note 4) | FAN_TACH_6 (Server Only) | GPI | None | |
| **GPP_G7** | None | Yes (Note 7) | Yes (Note 4) | FAN_TACH_7 (Server Only) | GPI | None | |
| **GPP_G8** | None | No | Yes (Note 4) | FAN_PWM_0 (Server Only) | FAN_PWM_0 | None | |

## Table 18-2. General Purpose I/O Signals (Sheet 9 of 12)

| Name | Internal Pull-up/ Pull-down (Note 1) | De-Glitch | | Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 2) | Default | NMI or SMI Capable | Note |
|---|---|---|---|---|---|---|---|
| | | Input | Output | | | | |
| **GPP_G9** | None | No | Yes (Note 4) | FAN_PWM_1 (Server Only) | FAN_PWM_1 | None | |
| **GPP_G10** | None | No | Yes (Note 4) | FAN_PWM_2 (Server Only) | FAN_PWM_2 | None | |
| **GPP_G11** | None | No | Yes (Note 4) | FAN_PWM_3 (Server Only) | FAN_PWM_3 | None | |
| **GPP_G12** | None | No | Yes (Note 4) | GSXDOUT | GPI | None | |
| **GPP_G13** | None | No | Yes (Note 4) | GSXSLOAD | GPI | None | |
| **GPP_G14** | None | No | Yes (Note 4) | GSXDIN | GPI | None | |
| **GPP_G15** | None | No | Yes (Note 4) | GSXRESET# | GPI | None | |
| **GPP_G16** | None | No | Yes (Note 4) | GSXCLK | GPI | None | |
| **GPP_G17** | None | No | Yes (Note 4) | ADR_COMPLETE | GPI | None | (Server use only. See Note 9) |
| **GPP_G18** | None | No | Yes (Note 4) | NMI# | GPI | None | Server Use only |
| **GPP_G19** | None | No | Yes (Note 4) | SMI# | GPI | None | Server use only |
| **GPP_G20** | None | No | Yes (Note 4) | None | GPI | None | |
| **GPP_G21** | None | No | Yes (Note 4) | None | GPI | None | |
| **GPP_G22** | None | No | Yes (Note 4) | None | GPI | None | |
| **GPP_G23** | None | No | Yes (Note 4) | None | GPI | None | |
| **Group H GPIO - Primary Power Well (1.8V or 3.3V)** | | | | | | | |
| **GPP_H0** | None | No | Yes (Note 4) | SRCCLKREQ6# | GPI | None | |
| **GPP_H1** | None | No | Yes (Note 4) | SRCCLKREQ7# | GPI | None | |
| **GPP_H2** | None | No | Yes (Note 4) | SRCCLKREQ8# | GPI | None | |
| **GPP_H3** | None | No | Yes (Note 4) | SRCCLKREQ9# | GPI | None | |
| **GPP_H4** | None | No | Yes (Note 4) | SRCCLKREQ10# | GPI | None | |
| **GPP_H5** | None | No | Yes (Note 4) | SRCCLKREQ11# | GPI | None | |
| **GPP_H6** | None | No | Yes (Note 4) | SRCCLKREQ12# | GPI | None | |
| **GPP_H7** | None | No | Yes (Note 4) | SRCCLKREQ13# | GPI | None | |

## Table 18-2. General Purpose I/O Signals (Sheet 10 of 12)

| Name | Internal Pull-up/ Pull-down (Note 1) | De-Glitch | | Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 2) | Default | NMI or SMI Capable | Note |
|---|---|---|---|---|---|---|---|
| | | Input | Output | | | | |
| **GPP_H8** | None | No | Yes (Note 4) | SRCCLKREQ14# | GPI | None | |
| **GPP_H9** | None | No | Yes (Note 4) | SRCCLKREQ15# | GPI | None | |
| **GPP_H10** | None | Yes (Note 7) | Yes (Note 4) | SML2CLK (Server Only) | GPI | None | |
| **GPP_H11** | None | Yes (Note 7) | Yes (Note 4) | SML2DATA (Server Only) | GPI | None | |
| **GPP_H12** | 20K PD (Note 3 | Yes (Note 7) | No | SML2ALERT# (Server/WS Only) | GPO | None | •Also used as a strap.<br>•The pull-down resistor is disabled after RSMRST# de-asserts<br>•As GPO, the signal defaults to '0' |
| **GPP_H13** | None | Yes (Note 7) | Yes (Note 4) | SML3CLK (Server Only) | GPI | None | |
| **GPP_H14** | None | Yes (Note 7) | Yes (Note 4) | SML3DATA (Server Only) | GPI | None | |
| **GPP_H15** | None | Yes (Note 7) | Yes (Note 4) | SML3ALERT# (Server Only) | GPI | None | |
| **GPP_H16** | None | Yes (Note 7) | Yes (Note 4) | SML4CLK (Server Only) | GPI | None | |
| **GPP_H17** | None | Yes (Note 7) | Yes (Note 4) | SML4DATA (Server Only) | GPI | None | |
| **GPP_H18** | None | Yes (Note 7) | Yes (Note 4) | SML4ALERT# (Server Only) | GPI | None | |
| **GPP_H19** | None | Yes (Note 7) | Yes (Note 4) | ISH_I2C0_SDA | GPI | None | |
| **GPP_H20** | None | Yes (Note 7) | Yes (Note 4) | ISH_I2C0_SCL | GPI | None | |
| **GPP_H21** | None | Yes (Note 7) | Yes (Note 4) | ISH_I2C1_SDA | GPI | None | |
| **GPP_H22** | None | Yes (Note 7) | Yes (Note 4) | ISH_I2C1_SCL | GPI | None | |
| **GPP_H23** | None | No | Yes (Note 4) | None | GPI | None | |
| **Group I GPIO - Primary Power Well (3.3V Only)** | | | | | | | |
| **GPP_I0** | None | No | Yes (Note 4) | DDPB_HPD0 | GPI | NMI SMI | |
| **GPP_I1** | None | No | Yes (Note 4) | DDPC_HPD1 | GPI | NMI SMI | |
| **GPP_I2** | None | No | Yes (Note 4) | DDPD_HPD2 | GPI | NMI SMI | |
| **GPP_I3** | None | No | Yes (Note 4) | DDPE_HPD3 | GPI | NMI SMI | |
| **GPP_I4** | None | No | Yes (Note 4) | EDP_HPD | GPI | None | |

## Table 18-2. General Purpose I/O Signals (Sheet 11 of 12)

| Name | Internal Pull-up/ Pull-down (Note 1) | De-Glitch | | Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 2) | Default | NMI or SMI Capable | Note |
|---|---|---|---|---|---|---|---|
| | | Input | Output | | | | |
| **GPP_I5** | None | Yes (Note 7) | Yes (Note 4) | DDPB_CTRLCLK | GPI | None | |
| **GPP_I6** | 20K PD (Note 3) | Yes (Note 7) | No | DDPB_CTRLDATA | GPO | None | •Also used as a strap.<br>•The pull-down resistor is disabled after PCH_PWROK de-asserts<br>•As GPO, the signal defaults to '0' |
| **GPP_I7** | None | Yes (Note 7) | Yes (Note 4) | DDPC_CTRLCLK | GPI | None | |
| **GPP_I8** | 20K PD (Note 3) | Yes (Note 7) | No | DDPC_CTRLDATA | GPO | None | •Also used as a strap.<br>•The pull-down resistor is disabled after PCH_PWROK de-asserts<br>•As GPO, the signal defaults to '0' |
| **GPP_I9** | None | Yes (Note 7) | Yes (Note 4) | DDPD_CTRLCLK | GPI | None | |
| **GPP_I10** | 20K PD (Note 3) | Yes (Note 7) | No | DDPD_CTRLDATA | GPO | None | •Also used as a strap.<br>•The pull-down resistor is disabled after PCH_PWROK de-asserts<br>•As GPO, the signals defaults to '0' |
| **GPIO In Deep Sleep Power Well (3.3V Only)** | | | | | | | |
| **GPD0** | None | No | Yes (Note 4) | BATLOW# | BATLOW# | None | |
| **GPD1** | None | No | Yes (Note 4) | ACPRESENT | ACPRESENT | None | |
| **GPD2** | None | No | Yes (Note 4) | LAN_WAKE# | LAN_WAKE# | None | |
| **GPD3** | None | Yes (Note 7) | Yes (Note 4) | PWRBTN# | PWRBTN# | None | |
| **GPD4** | None | No | Yes (Note 5) | SLP_S3# | SLP_S3# | None | |
| **GPD5** | None | No | Yes (Note 5) | SLP_S4# | SLP_S4# | None | |
| **GPD6** | None | No | Yes (Note 5) | SLP_A# | SLP_A# | None | |
| **GPD7** | None | No | Yes (Note 5) | Reserved Functionality | Reserved Functionality (Needs to be programmed for GPIO) | None | The reserved functionality defaults to an output. During reset, the signal is low and right after reset it's high by default. |
| **GPD8** | None | No | Yes (Note 5) | SUSCLK | SUSCLK | None | |
| **GPD9** | None | No | Yes (Note 5) | SLP_WLAN# | SLP_WLAN# | None | |
| **GPD10** | None | No | Yes (Note 5) | SLP_S5# | SLP_S5# | None | |

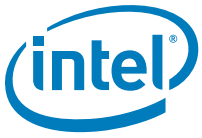


**Table 18-2. General Purpose I/O Signals (Sheet 12 of 12)**

| Name | Internal Pull-up/ Pull-down (Note 1) | De-Glitch | | Multiplexed With (1st = First Native Function 2nd = Second Native Function 3rd = Third Native Function) (Note 2) | Default | NMI or SMI Capable | Note |
|---|---|---|---|---|---|---|---|
| | | Input | Output | | | | |
| **GPD11** | None | No | Yes (Note 5) | LANPHYPC | LANPHYPC | None | |

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

# 18.5 Integrated Pull-ups and Pull-downs

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

See Table 18-2 (Internal Pull-up / Pull-down) for info on which GPIO has integrated PU/PD enabled by default.

# 18.6 Functional Description

## 18.6.1 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]

## 18.6.2 Blink/PWM Capability

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

To enable blink/PWM capability, the Pad Mode (in PAD_CFG_DW0_GPD_x register) of the corresponding GPIO needs to be set to 04h. See the Datasheet Volume 2 for more info on the register.

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

The PWM output is controlled by two different settings:

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

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

$$OutputFrequency = pwm_{clk} * \frac{base_unit_value}{256}$$

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

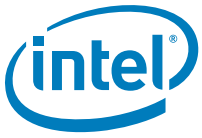
**Table 18-3. PWM Output Frequencies Assuming 32.768 KHz**

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

### 18.6.2.1 PWM Programming Sequence

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

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



### 18.6.3 Triggering

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

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

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

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

### 18.6.4 Sx GPIO Implementation Considerations

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

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

| GPIO Pin Configuration | Recommendation | Comment |
|---|---|---|
| Defaults to GPIO and used as GPI | Option 1: BIOS configures PADRSTCFG to select PLTRST#.<br>Option 2: BIOS disables RX path via GPIORXDIS bit and disables any enabled pull-up resistor via TERM bit prior to Sx entry.<br>***Note:*** BIOS needs to ensure GPIO resistors are restored appropriately when resuming from Sx. | With option 1, when PLTRST# asserts upon Sx entry, GPIORXDIS register bit defaults to ‘1’, which blocks its RX path. Other register bits in PAD_CFG_DW0_GPP_x and PAD_CFG_DW1_GPP_x will be also reset to default values. |
| Defaults to GPIO and used as GPO | Option 1: BIOS configures PADRSTCFG to select PLTRST#.<br>Option 2: BIOS disables TX path via GPIOTXDIS bit and disables any enabled pull-up resistor via TERM bit prior to Sx entry.<br>***Note:*** BIOS needs to ensure GPIO resistors are restored appropriately when resuming from Sx. | With option 1, when PLTRST# upon Sx entry, asserts, GPIOTXDIS register bit defaults to ‘1’, which blocks its TX path. Other register bits in PAD_CFG_DW0_GPP_x and PAD_CFG_DW1_GPP_x will be also reset to default values. |

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

## 18.6.5 GPIO Ownership

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

## 18.6.6 GPIO Pad Voltage Tolerance Configuration

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

- Group C: GPP_C[19:16]
- Group D: GPP_D4 and GPP_D23
- Group H: GPP_H[9:0] and GPP_H[22:19].

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

§ §

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

## 19.1 Acronyms

| Acronyms | Description |
|---|---|
| GSPI | Generic Serial Peripheral Interface |
| LTR | Latency Tolerance Reporting |

## 19.2 References

None

## 19.3 Overview

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

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

## 19.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **GSPI0_CS#/** GPP_B15 | O | **Generic SPI 0 Chip Select** |
| **GSPI0_CLK/** GPP_B16 | O | **Generic SPI 0 Clock** |
| **GSPI0_MISO/** GPP_B17 | I | **Generic SPI 0 MISO** |
| **GSPI0_MOSI/** GPP_B18 | O | **Generic SPI 0 MOSI**<br>***Note:*** This signal is also utilized as a strap. See the pin strap section for more info. |
| **GSPI1_CS#/** GPP_B19 | O | **Generic SPI 1 Chip Select** |
| **GSPI1_CLK/** GPP_B20 | O | **Generic SPI 1 Clock** |
| **GSPI1_MISO/** GPP_B21 | I | **Generic SPI 1 MISO** |
| **GSPI1_MOSI/** GPP_B22 | O | **Generic SPI 1 MOSI**<br>***Note:*** This signal is also utilized as a strap. See the pin strap section for more info. |

## 19.5 Integrated Pull-Ups and Pull-Downs

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

## 19.6 I/O Signal Planes and States

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

## 19.7 Functional Description

### 19.7.1 Features

The GSPI interfaces support the following features:

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

***Note:*** Slave mode is not supported.

### 19.7.2 Controller Overview

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

The processor or DMA accesses data through the transmit and receive FIFOs.

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

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

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

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

The GSPI controller, as a 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.

## 19.7.3 DMA Controller

The GSPI controllers have an integrated DMA controller.

### 19.7.3.1 DMA Transfer and Setup Modes

The DMA can operate in the following modes:

1. Memory to peripheral transfers. This mode requires 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.

### 19.7.3.2 Channel Control

- The source transfer width and destination transfer width are programmed. The width can be programmed to 1, 2, or 4 bytes.
- Burst size is configurable per channel for source and destination. The number is a power of 2 and can vary between 1,2,4,...,128. This number times the transaction width gives the number of bytes that will be transferred per burst.
- Individual Channel enables. If the channel is not being used, then it should be clock gated.
- Programmable Block size and Packing/Unpacking. Block size of the transfer is programmable in bytes. The block size is not limited by the source or destination transfer widths.

- Address incrementing modes: The DMA has a configurable mechanism for computing the source and destination addresses for the next transfer within the current block. The DMA supports incrementing addresses and constant addresses.
- Flexibility to configure any hardware handshake sideband interface to any of the DMA channels.
- Early termination of a transfer on a particular channel.

## 19.7.4 Reset

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

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

## 19.7.5 Power Management

### 19.7.5.1 Device Power Down Support

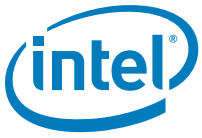
To power down peripherals connected to the PCH GSPI bus, the idle configured state of the I/O signals must be retained to avoid transitions on the bus that can affect the connected powered peripheral. Connected devices are allowed to remain in the D0 active or D2 low power states when the bus is powered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

### 19.7.5.2 Latency Tolerance Reporting (LTR)

Latency Tolerance Reporting is used to allow the system to optimize internal power states based on dynamic data, comprehending the current platform activity and service latency requirements. 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.



### 19.7.6 Interrupts

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

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

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

### 19.7.7 Error Handling

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

§ §

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

## 20.1 Acronyms

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

## 20.2 References

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

## 20.3 Overview

The PCH implements four I2C controllers for four independent I2C interfaces, I2C0-I2C3. Each interface is a two-wire serial interface consisting of a serial data line (SDA) and a serial clock (SCL).

## 20.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **I2C0_SDA**/ GPP_C16 | I/OD | **I2C Link 0 Serial Data Line** External Pull-up required. |
| **I2C0_SCL**/ GPP_C17 | I/OD | **I2C Link 0 Serial Clock Line** External Pull-up required. |
| **I2C1_SDA**/ GPP_C18 | I/OD | **I2C Link 1 Serial Data Line** External Pull-up required. |
| **I2C1_SCL**/ GPP_C19 | I/OD | **I2C Link 1 Serial Clock Line** External Pull-up required. |
| **I2C2_SDA**/ GPP_D13/ ISH_UART0_RXD/ SML0BDATA | I/OD | **I2C Link 2 Serial Data Line** External Pull-up required. |
| **I2C2_SCL**/ GPP_D14/ ISH_UART0_TXD/ SML0BCLK | I/OD | **I2C Link 2 Serial Clock Line** External Pull-up required. |

| Name | Type | Description |
|---|---|---|
| **I2C3_SDA**/ SH_I2C2_SDA / GPP_D4 | I/OD | **I2C Link 3 Serial Data Line** External Pull-up required. |
| **I2C3_SCL**/ SH_I2C2_SCL / GPP_D23 | I/OD | **I2C Link 3 Serial Clock Line** External Pull-up required. |

## 20.5 Integrated Pull-Ups and Pull-Downs

None

## 20.6 I/O Signal Planes and States

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| **I2C[3:0]_SDA** | Primary | Undriven | Undriven | Undriven | Off |
| **I2C[3:0]_SCL** | Primary | Undriven | Undriven | Undriven | Off |

## 20.7 Functional Description

### 20.7.1 Features

The I2C interfaces support the following features:

- Speed: standard mode (up to 100 Kb/s), fast mode (up to 400 Kb/s), and fast mode plus (up to 1 MB/s)
- 1.8V or 3.3V support (depending on the voltage supplied to the I2C signal group)
- Master I2C operation only
- 7-bit or 10-bit addressing
- 7-bit or 10-bit combined format transfers
- Bulk transmit mode
- Ignoring CBUS addresses (an older ancestor of I2C used to share the I2C bus)
- Interrupt or polled-mode operation
- Bit and byte waiting at all bus speed
- Component parameters for configurable software driver support
- Programmable SDA hold time ($t_{HD}$; DAT)
- DMA support with 64-byte DMA FIFO per channel (up to 32-byte burst)
- 64-byte Tx FIFO and 64-byte Rx FIFO
- SW controlled serial data line (SDA) and serial clock (SCL)

***Notes:***

1. High speed mode (up to 3.4 MB/s) is not supported.
2. The controllers must only be programmed to operate in master mode only. I2C slave mode is not supported.

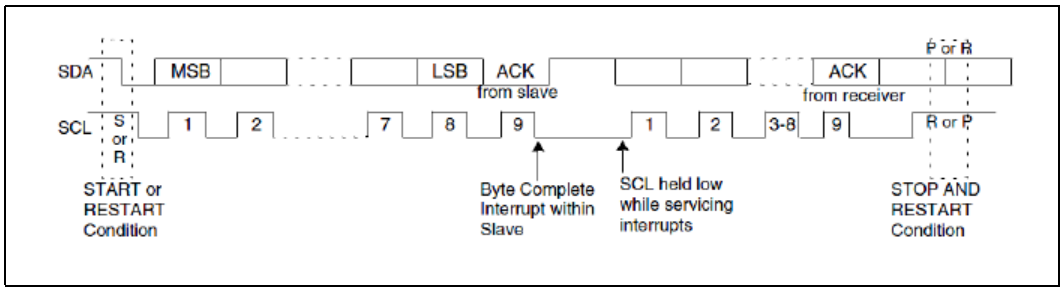
3. I2C multi masters are not supported.
4. Simultaneous configuration of Fast Mode and Fast Mode Plus is not supported.
5. I2C General Call is not supported.

## 20.7.2 Protocols Overview

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

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

**Figure 20-1. Data Transfer on the I2C Bus**




### 20.7.2.1 Combined Formats

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

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

To initiate combined format transfers, IC_CON.IC_RESTSART_EN should be set to 1. With this value set and operating as a master, when the controller completes an I2C transfer, it checks the transmit FIFO and executes the next transfer. If the direction of this transfer differs from the previous transfer, the combined format is used to issue the transfer. If the transmit FIFO is empty when the current I2C transfer completes, a STOP is issued and the next transfer is issued following a START condition.

## 20.7.3 DMA Controller

The $I^2C$ controllers 0 to 3 (I2C0 - I2C3) each has an integrated DMA controller.

### 20.7.3.1 DMA Transfer and Setup Modes

The DMA can operate in the following modes:

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

The DMA supports the following modes for programming:

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

### 20.7.3.2 Channel Control

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

## 20.7.4 Reset

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

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

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

## 20.7.5 Power Management

### 20.7.5.1 Device Power Down Support

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

### 20.7.5.2 Latency Tolerance Reporting (LTR)

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

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

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

## 20.7.6 Interrupts

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

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

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

## 20.7.7 Error Handling

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

## 20.7.8 Programmable SDA Hold Time

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

# 21 Gigabit Ethernet Controller

## 21.1 Acronyms

| Acronyms | Description |
|---|---|
| GbE | Gigabit Ethernet |

## 21.2 References

| Specification | Location |
|---|---|
| *Alert Standard Format Specification, Version 1.03* | http://www.dmtf.org/standards/asf |
| *IEEE 802.3 Fast Ethernet* | http://standards.ieee.org/getieee802/ |
| *Intel® Ethernet Connection I219 Datasheet* | TBD |

## 21.3 Overview

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

## 21.4 Signal Description

### Table 21-1. GbE LAN Signals (Sheet 1 of 2)

| Name | Type | Description |
|---|---|---|
| **PCIE4_TXP /** USB3_10_TXP<br>**PCIE4_TXN /** USB3_10_TXN<br>**PCIE5_TXP**<br>**PCIE5_TXN**<br>**PCIE9_TXP /** SATA0A_TXP<br>**PCIE9_TXN /** SATA0A_TXN<br>**PCIE12_TXP**<br>**PCIE12_TXN**<br>**PCIE13_TXP /** SATA0B_TXP<br>**PCIE13_TXN /** SATA0B_TXN | O | Refer to Chapter 26 for details on the PCI Express transmit signals.<br><br>***Note:*** The Intel® Ethernet Connection I219 can be connected to one of the following PCI Express ports 4, 5, 9, 12, 13. on PCH-H. |
| **PCIE4_RXP** / USB3_10_RXP<br>**PCIE4_RXN** / USB3_10_RXN<br>**PCIE5_RXP**<br>**PCIE5_RXN**<br>**PCIE9_RXP** / SATA0A_RXP<br>**PCIE9_RXN** / SATA0A_RXN<br>**PCIE12_RXP**<br>**PCIE12_RXN**<br>**PCIE13_RXP** / SATA0B_RXP<br>**PCIE13_RXN** / SATA0B_RXN | I | Refer to Chapter 26 for details on the PCI Express receive signals.<br><br>***Note:*** The Intel® Ethernet Connection I219 can be connected to one of the following PCI Express ports 4, 5, 9, 12, 13. on PCH-H. |

### Table 21-1. GbE LAN Signals (Sheet 2 of 2)

| Name | Type | Description |
|---|---|---|
| **SML0DATA/**GPP_C4 | I/OD | Refer to Chapter 30 for details on the SML0DATA signal.<br>***Note:*** The Intel® Ethernet Connection I219 connects to SML0DATA signal. |
| **SML0CLK/**GPP_C3 | I/OD | Refer to Chapter 30 for details on the SML0CLK signal.<br>***Note:*** The Intel® Ethernet Connection I219 connects to SML0CLK signal. |
| **LANPHYPC/**GPD11 | O | **LAN PHY Power Control:** LANPHYPC should be connected to LAN_DISABLE_N on the PHY. PCH will drive LANPHYPC low to put the PHY into a low power state when functionality is not needed.<br>***Note:*** LANPHYPC can only be driven low if SLP_LAN# is de-asserted.<br>***Note:*** Signal can instead be used as GPD11. |
| **SLP_LAN#** | O | **LAN Sub-System Sleep Control:** If the Gigabit Ethernet Controller is enabled, when SLP_LAN# is de-asserted it indicates that the PHY device must be powered. When SLP_LAN# is asserted, power can be shut off to the PHY device. SLP_LAN# will always be de-asserted in S0 and anytime SLP_A# is de-asserted<br><br>***Note:*** If Gigabit Ethernet Controller is statically disabled via soft-strap or BIOS, SLP_LAN# will be driven low. |
| **LAN_WAKE#/**GPD2 | I | **LAN WAKE:** LAN Wake Indicator from the GbE PHY.<br><br>***Note:*** Signal can instead be used as GPD2. |

## 21.5 Integrated Pull-Ups and Pull-Downs

### Table 21-2. Integrated Pull-Ups and Pull-Downs

| Signal | Resistor Type | Value Ω | Notes |
|---|---|---|---|
| **LAN_WAKE#/**GPD2 | External Pull-up required. Internal Pull-down may be enabled in DeepSx | 15k-40k | |

## 21.6 I/O Signal Planes and States

### Table 21-3. Power Plane and States for Output Signals

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

***Note:***
1. Based on wake events and Intel ME state

### Table 21-4. Power Plane and States for Input Signals (Sheet 1 of 2)

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| **LAN_WAKE#/** GPD2 | DSW | Undriven | Undriven | Undriven | Undriven/ Internal Pull-down[2] |

**Table 21-4. Power Plane and States for Input Signals (Sheet 2 of 2)**

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| ***Notes:***<br>1. Configurable<br>2. Configurable based on PMC configuration bit.<br>'1' (pin will be driven by platform in DeepSx) -> Undriven;<br>'0' (pin will NOT be driven by platform in DeepSx) -> Internal Pull-down (15k-40k) enabled | | | | | |

# 21.7 Functional Description

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

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

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

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

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

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

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

The integrated GbE controller features are:

- Network Features
  - — Compliant with the 1 GB/s Ethernet 802.3, 802.3u, 802.3ab specifications
  - — Multi-speed operation: 10/100/1000 Mbps
  - — Full-duplex operation at 10/100/1000 Mbps: Half-duplex at 10/100 Mbps
  - — Flow control support compliant with the 802.3X specification
  - — VLAN support compliant with the 802.3q specification
  - — MAC address filters: perfect match unicast filters; multicast hash filtering, broadcast filter and promiscuous mode
  - — PCI Express/SMLink interface to GbE PHYs

- Host Interface Features
  - 64-bit address master support for systems using more than 4 GB of physical memory
  - Programmable host memory receive buffers (256 bytes to 16 KB)
  - Intelligent interrupt generation features to enhance driver performance
  - Descriptor ring management hardware for transmit and receive
  - Software controlled reset (resets everything except the configuration space)
  - Message Signaled Interrupts
- Performance Features
  - Configurable receive and transmit data FIFO, programmable in 1 KB increments
  - TCP segmentation off loading features
  - Fragmented UDP checksum off load for packet reassembly
  - IPv4 and IPv6 checksum off load support (receive, transmit, and large send)
  - Split header support to eliminate payload copy from user space to host space
  - Receive Side Scaling (RSS) with two hardware receive queues
  - Supports 9018 bytes of jumbo packets
  - Packet buffer size 32 KB
  - TimeSync off load compliant with 802.1as specification
  - Platform time synchronization
- Power Management Features
  - Magic Packet* wake-up enable with unique MAC address
  - ACPI register set and power down functionality supporting D0 and D3 states
  - Full wake up support (APM, ACPI)
  - MAC power down at Sx, DM-Off with and without WoL
  - Auto connect battery saver at S0 no link and Sx no link
  - Energy Efficient Ethernet (EEE) support
  - Latency Tolerance Reporting (LTR)
  - ARP and ND proxy support through LAN Connected Device proxy
  - Wake on LAN (WoL) from Deep Sx
  - Windows* InstantGo* Support

## 21.7.1 GbE PCI Express* Bus Interface

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

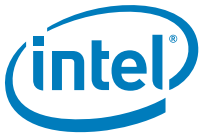
### 21.7.1.1 Transaction Layer

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

### 21.7.1.2 Data Alignment

#### 21.7.1.2.1 4-KB Boundary

PCI requests must never specify an address/length combination that causes a memory space access to cross a 4-KB boundary. It is hardware's responsibility to break requests into 4-KB aligned requests (if needed). This does not pose any requirement on



software. However, if software allocates a buffer across a 4-KB boundary, hardware issues multiple requests for the buffer. Software should consider aligning buffers to a 4-KB boundary in cases where it improves performance. The alignment to the 4-KB boundaries is done by the GbE controller. The transaction layer does not do any alignment according to these boundaries.

#### 21.7.1.2.2 PCI Request Size

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

### 21.7.1.3 Configuration Request Retry Status

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

## 21.7.2 Error Events and Error Reporting

### 21.7.2.1 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#.

### 21.7.2.2 Unsupported Request Error Handling

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

## 21.7.3 Ethernet Interface

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

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

#### 21.7.3.1 Intel® Ethernet Connection I219

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

The integrated GbE controller supports various modes as listed in Table 21-5.

**Table 21-5. LAN Mode Support**

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

### 21.7.4 PCI Power Management

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

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

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

§ §

# 22 Interrupt Interface

## 22.1 Acronyms

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

## 22.2 References

None

## 22.3 Overview

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

## 22.4 Signal Description

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

## 22.5 Integrated Pull-Ups and Pull-Downs

None

## 22.6 I/O Signal Planes and States

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

## 22.7 Functional Description

The PCH supports both APIC and PIC modes.

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

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

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

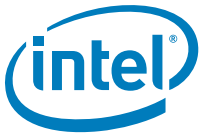
1. For multi-cause Initiators (Multiple Interrupt Causes from Single Source and Single SB Port ID, i.e. GPIO, eSPI): If more than 1 interrupt cause has to use the same IRQ#, it has to be aggregated or guaranteed through BIOS/SW to assign a unique IRQ per Interrupt Cause.
2. For multi-function devices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

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

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

**Table 22-1. Interrupt Options - 8259 Mode (Sheet 1 of 2)**

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



## Table 22-1. Interrupt Options - 8259 Mode (Sheet 2 of 2)

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

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

## Table 22-2. Interrupt Options - APIC Mode (Sheet 1 of 2)

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

### Table 22-2. Interrupt Options - APIC Mode (Sheet 2 of 2)

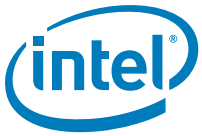
| IRQ# | Pin | SERIRQ | PCI Message | IRQ Sharable? | Internal Modules |
|---|---|---|---|---|---|
| 17:19 | No | PIRQ[B-D] | Yes | Yes | Option for configurable sources including internal PIRQ[B-D], GPIO, eSPI and internal ACPI/PCI devices |
| 20:23 | No | No | No | Yes | Option for configurable sources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

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

The following signals are associated with the Interrupt Logic.

### Table 22-3. Interrupt Logic Signals

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



## 22.7.1 8259 Interrupt Controllers (PIC)

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

**Table 22-4. 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. |
| 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#.

## 22.7.2 Interrupt Handling

### 22.7.2.1 Generating Interrupts

The PIC interrupt sequence involves three bits, from the IRR, ISR, and IMR, for each interrupt level. These bits are used to determine the interrupt vector returned, and status of any other pending interrupts. Table 22-5 defines the IRR, ISR, and IMR.

**Table 22-5. Interrupt Status Registers**

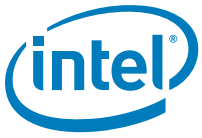
| Bit | Description |
|---|---|
| IRR | **Interrupt Request Register.** This bit is set on a low to high transition of the interrupt line in edge mode, and by an active high level in level mode. This bit is set whether or not the interrupt is masked. However, a masked interrupt will not generate INTR.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

### 22.7.2.2 Acknowledging Interrupts

The processor generates an interrupt acknowledge cycle that is translated by the host bridge into a PCI Interrupt Acknowledge Cycle to the PCH. The PIC translates this command into two internal INTA# pulses expected by the 8259 cores. The PIC uses the first internal INTA# pulse to freeze the state of the interrupts for priority resolution. On the second INTA# pulse, the 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 22-6. 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 |



#### 22.7.2.3 Hardware/Software Interrupt Sequence

1. One or more of the Interrupt Request lines (IRQ) are raised high in edge mode, or seen high in level mode, setting the corresponding IRR bit.
2. The PIC sends INTR active to the processor if an asserted interrupt is not masked.
3. The processor acknowledges the INTR and responds with an interrupt acknowledge cycle. The cycle is translated into a PCI interrupt acknowledge cycle by the host bridge. This command is broadcast over PCI by the PCH.
4. Upon observing its own interrupt acknowledge cycle on PCI, the PCH converts it into the two cycles that the internal 8259 pair can respond to. Each cycle appears as an interrupt acknowledge pulse on the internal INTA# pin of the cascaded interrupt controllers.
5. Upon receiving the first internally generated INTA# pulse, the highest priority ISR bit is set and the corresponding IRR bit is reset. On the trailing edge of the first pulse, a 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.

### 22.7.3 Initialization Command Words (ICWx)

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

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

#### 22.7.3.1 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 bytes 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.

#### 22.7.3.2 ICW2

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

#### 22.7.3.3 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.

#### 22.7.3.4 ICW4

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

### 22.7.4 Operation Command Words (OCW)

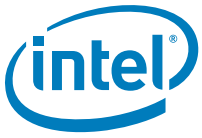
These command words reprogram the interrupt controller to operate in various interrupt modes.

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

### 22.7.5 Modes of Operation

#### 22.7.5.1 Fully-Nested Mode

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



### 22.7.5.2 Special Fully-Nested Mode

This mode is used in the case of a system where cascading is used, and the priority has to be conserved within each 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.

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

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

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

### 22.7.5.4 Specific Rotation Mode (Specific Priority)

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

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

### 22.7.5.5 Poll Mode

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

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

#### 22.7.5.6 Edge and Level Triggered Mode

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

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

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

#### 22.7.5.7 End Of Interrupt (EOI) Operations

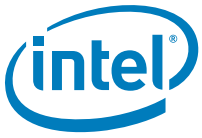
An EOI can occur in one of two fashions: by a command word writes issued to the PIC before returning from a service routine, the EOI command; or automatically when AEOI bit in ICW4 is set to 1.

#### 22.7.5.8 Normal End of Interrupt

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

#### 22.7.5.9 Automatic End of Interrupt Mode

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



### 22.7.6 Masking Interrupts

#### 22.7.6.1 Masking on an Individual Interrupt Request

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

#### 22.7.6.2 Special Mask Mode

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

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

### 22.7.7 Steering PCI Interrupts

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

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

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

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

### 22.8.1 Interrupt Handling

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

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

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

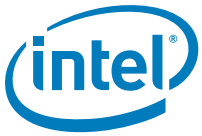
## 22.8.2 Interrupt Mapping

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

**Table 22-7. APIC Interrupt Mapping[1]**

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

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



### 22.8.3 PCI/PCI Express* Message-Based Interrupts

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

### 22.8.4 IOxAPIC Address Remapping

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

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

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

### 22.8.5 External Interrupt Controller Support

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

## 22.9 Serial Interrupt

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

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

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

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

### 22.9.1 Start Frame

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

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

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

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

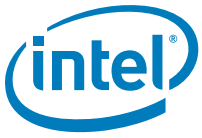
- **Sample Phase—**During this phase, the SERIRQ device drives SERIRQ low if the corresponding interrupt signal is low. If the corresponding interrupt is high, then the SERIRQ devices tri-state the SERIRQ signal. The SERIRQ line remains high due to Pull-up resistors (there is no internal Pull-up resistor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

### 22.9.2 Stop Frame

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

**Table 22-8. Stop Frame Explanation**

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



## 22.9.3 Specific Interrupts Not Supported Using SERIRQ

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

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

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

Table 22-9 shows the format of the data frames. For the PCI interrupts (A–D), the output from the PCH is AND'd with the PCI input signal. This way, the interrupt can be signaled using both the PCI interrupt input signal and using the SERIRQ signal (they are shared).

**Table 22-9. Data Frame Format**

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

§ §

# 23 Integrated Sensor Hub (ISH)

## 23.1 Acronyms

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

## 23.2 References

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

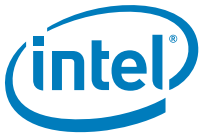
## 23.3 Overview

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

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

The ISH consists of the following key components:

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



## 23.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **ISH_I2C0_SDA/**GPP_H19 | I/OD | **$I^2C$ 0 Data** |
| **ISH_I2C0_SCL/**GPP_H20 | I/OD | **$I^2C$ 0 Clk** |
| **ISH_I2C1_SDA/**GPP_H21 | I/OD | **$I^2C$ 1 Data** |
| **ISH_I2C1_SCL/**GPP_H22 | I/OD | **$I^2C$ 1 Clk** |
| **ISH_I2C2_SDA** /GPP_D4 /I2C3_SDA | I/OD | **$I^2C$ 2 Data** |
| **ISH_I2C2_SCL /**GPP_D23 / I2C3_SCL | I/OD | **$I^2C$ 2 Clk** |
| **ISH_GP0/**GPP_A18 | I/O | **ISH GPIO 0** |
| **ISH_GP1/**GPP_A19 | I/O | **ISH GPIO 1** |
| **ISH_GP2/**GPP_A20 | I/O | **ISH GPIO 2** |
| **ISH_GP3/**GPP_A21 | I/O | **ISH GPIO 3** |
| **ISH_GP4/**GPP_A22 | I/O | **ISH GPIO 4** |
| **ISH_GP5/**GPP_A23 | I/O | **ISH GPIO 5** |
| **ISH_GP6/**BM_BUSY#/SX_EXIT_HOLDOFF# /GPP_A12 | I/O | **ISH GPIO 6** |
| **ISH_GP7/**GPP_A17 | I/O | **ISH GPIO 7** |
| **ISH_UART0_TXD** / GPP_D14 / SML0BCLK / I2C2_SCL | O | **UART 0 Transmit Data** |
| **ISH_UART0_RXD** /GPP_D13/SML0BDATA/I2C2_SDA | I | **UART 0 Receive Data** |
| **ISH_UART0_RTS#/**GPP_D15 | O | **UART 0 Request To Send** |
| **ISH_UART0_CTS#/**GPP_D16 | I | **UART 0 Clear to Send** |
| **ISH_UART1_TXD/**UART1_TXD/GPP_C13 | O | **UART 1 Transmit Data** |
| **ISH_UART1_RXD/**UART1_RXD/GPP_C12 | I | **UART 1 Receive Data** |
| **ISH_UART1_RTS#/**UART1_RTS#/GPP_C14 | O | **UART 1 Request To Send** |
| **ISH_UART1_CTS#/**UART1_CTS#/GPP_C15 | I | **UART 1 Clear to Send** |

## 23.5 Integrated Pull-Ups and Pull-Downs

None

## 23.6 I/O Signal Planes and States

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| **ISH_I2C0_SDA** | Primary | Undriven | Undriven | Undriven | OFF |
| **ISH_I2C0_SCL** | Primary | Undriven | Undriven | Undriven | OFF |
| **ISH_I2C1_SDA** | Primary | Undriven | Undriven | Undriven | OFF |
| **ISH_I2C1_SCL** | Primary | Undriven | Undriven | Undriven | OFF |
| **ISH_I2C2_SDA** | Primary | Undriven | Undriven | Undriven | OFF |
| **ISH_I2C2_SCL** | Primary | Undriven | Undriven | Undriven | OFF |
| **ISH_GP[7:0]** | Primary | Undriven | Undriven | Undriven | OFF |
| **ISH_UART0_TXD** | Primary | Undriven | Undriven | Undriven | OFF |

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| **ISH_UART0_RXD** | Primary | Undriven | Undriven | Undriven | OFF |
| **ISH_UART0_RTS#** | Primary | Undriven | Undriven | Undriven | OFF |
| **ISH_UART0_CTS#** | Primary | Undriven | Undriven | Undriven | OFF |
| **ISH_UART1_TXD** | Primary | Undriven | Undriven | Undriven | OFF |
| **ISH_UART1_RXD** | Primary | Undriven | Undriven | Undriven | OFF |
| **ISH_UART1_RTS#** | Primary | Undriven | Undriven | Undriven | OFF |
| **ISH_UART1_CTS#** | Primary | Undriven | Undriven | Undriven | OFF |

# 23.7 Functional Description

## 23.7.1 ISH Micro-Controller

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

## 23.7.2 SRAM

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

## 23.7.3 PCI Host Interface

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

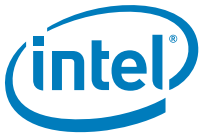
### 23.7.3.1 MMIO Space

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

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

### 23.7.3.2 DMA Controller

The DMA controller supports up to 64-bit addressing.



#### 23.7.3.3 PCI Interrupts

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

#### 23.7.3.4 PCI Power Management

PME is not supported in ISH.

### 23.7.4 Power Domains and Management

#### 23.7.4.1 ISH Power Management

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

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

#### 23.7.4.2 External Sensor Power Management

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

### 23.7.5 ISH IPC

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

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

**Table 23-1. IPC Initiator -> Target flows**

| Initiator | Target |
|---|---|
| ISH | Host processor |
| Host processor | ISH |
| ISH | Intel® ME |
| Intel® ME | ISH |

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

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

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

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

### 23.7.7 ISH I2C Controllers

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

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

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

### 23.7.8 ISH UART Controller

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

The UART has the following Capabilities:

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

### 23.7.9 ISH GPIOs

The ISH support eight dedicated GPIOs.

## 23.8 Embedded Location (Comms Hub)

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

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

| Connectivity Ingredient | Ingredient Name | Embedded Location Usage |
|---|---|---|
| Wi-Fi | Snowfield Peak Wi-Fi | Indoor Location |

| Connectivity Ingredient | Ingredient Name | Embedded Location Usage |
|---|---|---|
| Discrete GNSS | CG2000 | Outdoor Location |
| WWAN | 726x | Cell ID - Used for improved outdoor and indoor location identification |
| Sensors | Sensors connected to ISH | Used to provide accurate platform location by taking into account the sensor data in conjunction with other connectivity ingredients like Wi-Fi, GNSS, and WWAN |

§ §

# 24 Low Pin Count (LPC)

## 24.1 Acronyms

| Acronyms | Description |
|---|---|
| LPC | Low Pin Count |

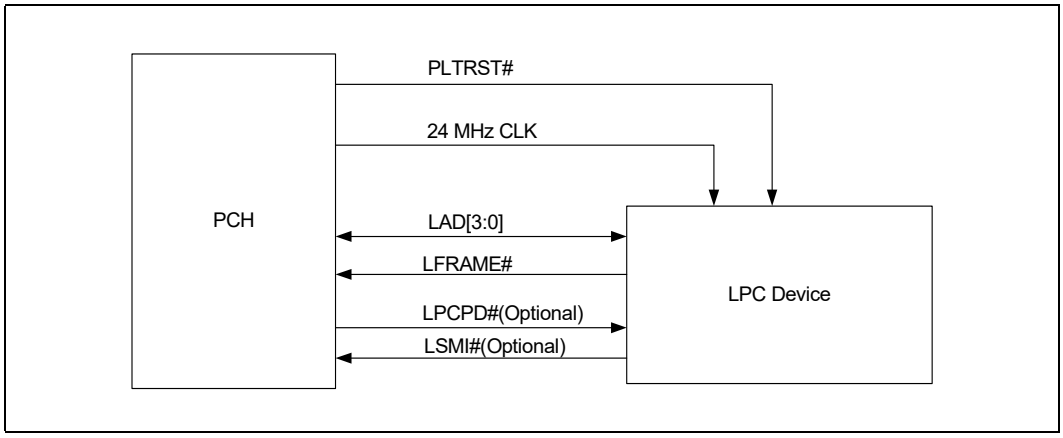
## 24.2 References

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

## 24.3 Overview

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

**Figure 24-1. LPC Interface Diagram**

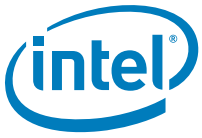



The PCH supports all of the signals that are shown as optional, but peripherals are not required to do so.

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

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

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



## 24.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **LAD0/** ESPI_IO0/ GPP_A1 | I/O | **LPC Multiplexed Command, Address, Data.** For LAD0, internal Pull-up is provided. |
| **LAD1/** ESPI_IO1/ GPP_A2 | I/O | **LPC Multiplexed Command, Address, Data.** For LAD1, internal Pull-up is provided. |
| **LAD2/** ESPI_IO2/ GPP_A3 | I/O | **LPC Multiplexed Command, Address, Data.** For LAD2, internal Pull-up is provided. |
| **LAD3/** ESPI_IO3/ GPP_A4 | I/O | **LPC Multiplexed Command, Address, Data.** For LAD3, internal Pull-up is provided. |
| **LFRAME#/** ESPI_CS#/ GPP_A5 | O | **LPC Frame:** LFRAME# indicates the start of an LPC cycle, or an abort. |
| **RCIN#/** GPP_A0 / ESPI ALERT1 (Server Only) | I | **Keyboard Controller Reset Processor:** The keyboard controller can generate INIT# to the processor. This saves the external OR gate with the PCH's other sources of INIT#. When the PCH detects the assertion of this signal, INIT# is generated to the processor.<br>***Note:*** The PCH will ignore RCIN# assertion during transitions to the S3, S4, and S5 states. |

## 24.5 Integrated Pull-Ups and Pull-Downs

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

## 24.6 I/O Signal Planes and States

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

## 24.7 Functional Description

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

### 24.7.1 LPC Cycle Types

The PCH implements the cycle types shown in Table 24-1.

**Table 24-1. LPC Cycle Types Supported (Sheet 1 of 2)**

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

**Table 24-1. LPC Cycle Types Supported (Sheet 2 of 2)**

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

***Notes:***

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

## 24.7.2 Start Field Definition

**Table 24-2. Start Field Bit Definitions**

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

***Note:*** All other encodings are RESERVED.

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

The PCH always drives Bit 0 of this field to 0. Table 24-3 shows the valid bit encodings.

**Table 24-3. Cycle Type Bit Definitions**

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

***Note:*** All other encodings are RESERVED.

## 24.7.4 Size

Bits[3:2] are reserved. The PCH always drives them to 00. Bits[1:0] are encoded as listed in Table 24-4.

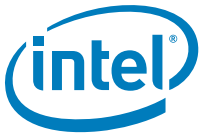


**Table 24-4. Transfer Size Bit Definition**

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

#### 24.7.4.1 SYNC

Valid values for the SYNC field are shown in Table 24-5.

## 24.7.5 SYNC Timeout

**Table 24-5. SYNC Bit Definition**

| Bits[3:0] | Indication |
|---|---|
| 0000 | **Ready:** SYNC achieved with no error. |
| 0101 | **Short Wait:** Part indicating wait-states. For bus master cycles, the PCH does not use this encoding. Instead, the PCH uses the Long Wait encoding (see next encoding below). |
| 0110 | **Long Wait:** Part indicating wait-states, and many wait-states will be added. This encoding driven by the PCH for bus master cycles, rather than the Short Wait (0101). |
| 1010 | **Error:** Sync achieved with error. This is generally used to replace the SERR# or IOCHK# signal on the PCI/ISA bus. It indicates that the data is to be transferred, but there is a serious error in this transfer. |
| ***Notes:***<br>1. All other combinations are RESERVED.<br>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.

## 24.7.6 SYNC Error Indication

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

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

## 24.7.7 LFRAME# Usage

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

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

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

## 24.7.8 I/O Cycles

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

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

## 24.7.9 LPC Power Management

### 24.7.9.1 LPCPD# Protocol

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

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

## 24.7.10 Configuration and PCH Implications

### 24.7.10.1 LPC I/F Decoders

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

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

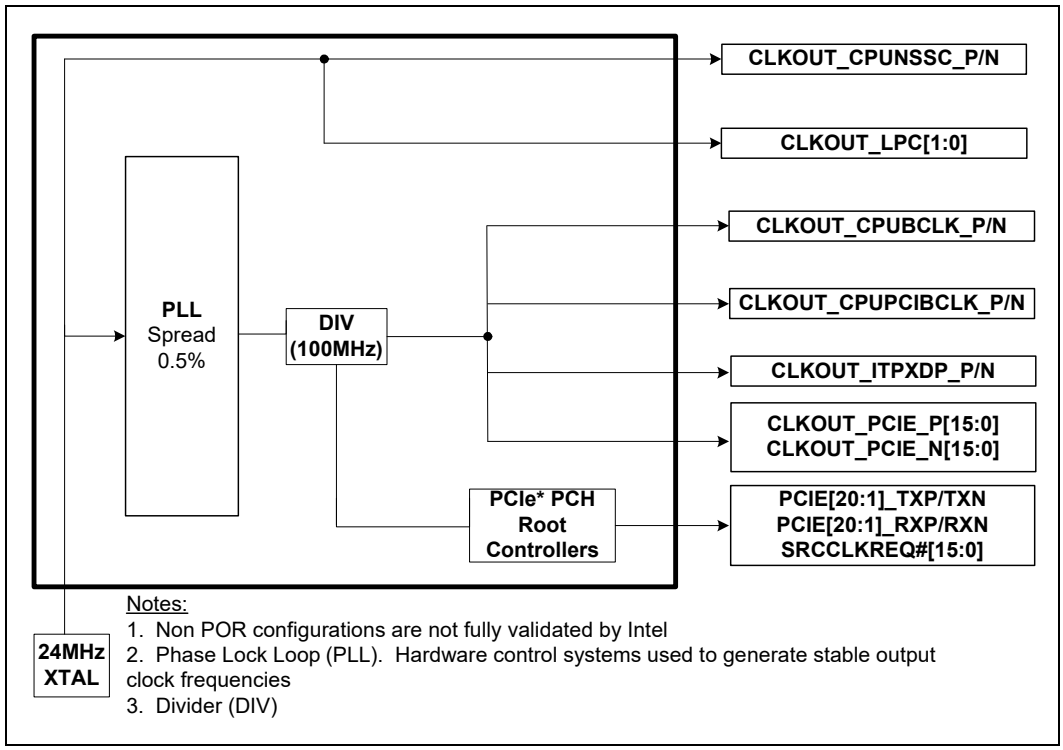
§ §

# 25 PCH and System Clocks

## 25.1 Overview

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

**Figure 25-1. PCH Internal Clock Diagram**




## 25.2 Signal Descriptions

| Name | Type | SSC Capable | Description |
|---|---|---|---|
| CLKOUT_ITPXDP_P<br>CLKOUT_ITPXDP_N | O | Yes | **Differential ITP Debug Clock:** 100-MHz differential output to processor XDP/ITP connector on the platform |
| CLKOUT_CPUNSSC_P<br>CLKOUT_CPUNSSC_N | O | No | **Unfiltered Clock from Crystal to CPU:** 24-MHz differential re-buffered crystal reference clock to the processor |
| CLKOUT_CPUPCIBCLK_P<br>CLKOUT_CPUPCIBCLK_N | O | Yes | **Differential PCIe* Reference Clock to CPU:** 100-MHz PCIe* 3.0 specification compliant differential PCIe* reference clock to the processor |
| CLKOUT_CPUBCLK_P<br>CLKOUT_CPUBCLK_N | O | Yes | **Differential Clock to CPU:** 100-MHz differential core reference clock to the processor |

| Name | Type | SSC Capable | Description |
|---|---|---|---|
| CLKOUT_PCIE_P[15:0]<br>CLKOUT_PCIE_N[15:0] | O | Yes | **PCI Express* Clock Output:** 100-MHz PCIe* 3.0 specification compliant differential output clocks to PCIe* devices |
| CLKOUT_LPC[1:0] | O | No | **Low Pin Count (LPC) Clock Outputs:** Single-Ended 24-MHz output to various single load connectors/devices |
| CLKOUT_48 | O | No | **48 Clock Output (PCH-H Server Only):** Single-Ended 48-MHz output to Server BMC devices |
| SRCCLKREQ#[15:0] | I/O | N/A | **Clock Request:** Clock request signals for PCIe* 100-MHz differential clocks |
| XTAL24_IN | I | N/A | **Crystal Input:** Input connection for 24-MHz crystal to PCH oscillator circuit |
| XTAL24_OUT | O | N/A | **Crystal Output:** Output connection for 24-MHz crystal to PCH oscillator circuit |
| XCLK_BIASREF | I/O | N/A | **Differential Clock Bias Reference:** Used to set BIAS reference for differential clocks |

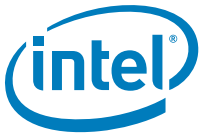
***Notes:***
1. SSC = Spread Spectrum Clocking. Intel does not recommend changing the Plan of Record and fully validated SSC default value set in BIOS Reference Code. The SSC level must only be adjusted for debugging or testing efforts and any Non POR configuration setting used are the sole responsibility of the customer.
2. N/A = Not Applicable
3. The SRCCLKREQ#[15:0] signals can be configured to map to any of the PCH-H PCI Express* Root Ports
4. SRCCLKREQ#[15:0] to CLKOUT_PCIE_P/N[15:0] Mapping Requirements
   - SRCCLKREQ#[7:0] signals can be mapped to any of the CLKOUT_PCIE_P/N[7:0] differential clock pairs
   - SRCCLKREQ#[15:8] signals can be mapped to any of the CLKOUT_PCIE_P/N[15:8] differential clock pairs

## 25.3 I/O Signal Planes and States

### Table 25-1. I/O Signal Planes and States

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| CLKOUT_ITPXDP_P<br>CLKOUT_ITPXDP_N | Primary | Toggling | Toggling | Driven Low | OFF |
| CLKOUT_CPUNSSC_P<br>CLKOUT_CPUNSSC_N | Primary | Toggling | Toggling | Driven Low | OFF |
| CLKOUT_CPUPCIBCLK_P<br>CLKOUT_CPUPCIBCLK_N | Primary | Toggling | Toggling | Driven Low | OFF |
| CLKOUT_CPUBCLK_P<br>CLKOUT_CPUBCLK_P | Primary | Toggling | Toggling | Driven Low | OFF |
| CLKOUT_PCIE_P[15:0]<br>CLKOUT_PCIE_N[15:0] | Primary | Toggling | Toggling | Driven Low | OFF |
| CLKOUT_LPC[1:0] | Primary | Toggling | Toggling | Driven Low | OFF |
| CLKOUT_48 | Primary | Toggling | Toggling | Driven Low | OFF |
| SRCCLKREQ#[15:0] | Primary | Un-driven | Un-driven | Un-driven | OFF |
| XTAL24_IN | Primary | Un-driven | Un-driven | Un-driven | OFF |
| XTAL24_OUT | Primary | Un-driven | Un-driven | Un-driven | OFF |
| XCLK_BIASREF | Primary | Un-driven | Un-driven | Un-driven | OFF |

***Notes:***
1. CLKOUT_48 is only supported and enabled on PCH-H Server



## 25.4 General Features

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

§ §

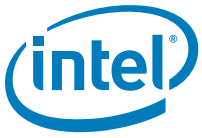
# 26 PCI Express* (PCIe*)

## 26.1 References

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

## 26.2 Overview

- PCH-H supports up to 16 PCIe* Ports and 20 PCIe* Lanes, with transfer rates up to 8 GT/s (Gen3)
- PCI Express* Gen 1 and Gen 2 ExpressCard 1.0 module-based hot-plug support
- Dynamic Link Throttling
- Port 8xh Decode
- PCI Express* Gen 1 and Gen 2 Separate Reference Clock with Independent Spread Spectrum Clocking (SRIS) Support
- Latency Tolerance Reporting
- End-to-End PCI Express* Controller Lane Reversal
- Access Control Services
- Alternative Routing ID
- Autonomous Link Width Negotiation as a target
- Advanced Error Reporting
- PCI Express* Lane Polarity Inversion
- Configurable 128B or 256B Maximum Data Payload
- PCIe* Subtractive Decode is not supported
  - PCI can still be supported via a PCIe*-to-PCI bridge. However, legacy PCI devices (such as PCMCIA or non-plug-and-play device) that need subtractive decode are not supported.
- Intel® Rapid Storage Technology (Intel® RST) for PCIe* Gen 1, Gen 2, and Gen 3 Storage Support
- PCI Express* Gen 1 and Gen 2 Receiver (RX) L0s Link Power Management State Support
- PCI Express* Gen 1 and Gen 2 External Graphics Support
- Single-Root I/O Virtualization (SR-IOV) Alternative Routing-ID Interpretation (ARI) and Access Control Services (ACS) feature support



## 26.3 Signal Description

| PCH | Name | Type | Description |
|---|---|---|---|
| PCH-H | PCIE[20:1]_TXP<br>PCIE[20:1]_TXN | O | **PCI Express* Differential Transmit Pairs 1 to 20**<br>These are PCI Express* based outbound high-speed differential signals |
| | PCIE[20:1]_RXP<br>PCIE[20:1]_RXN | I | **PCI Express* Differential Receive Pairs 1 to 20**<br>These are PCI Express* based inbound high-speed differential signals |
| | PCIE_RCOMPP<br>PCIE_RCOMPN | I | **Impedance Compensation Inputs** |

## 26.4 I/O Signal Planes and States

| Signal Name | Type | Power Plane | During Reset | Immediately After Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|---|
| PCIE[20:1]_TXP<br>PCIE[20:1]_TXN | O | Primary | Internal Pull-down | Internal Pull-down | Internal Pull-down | Off |
| PCIE[20:1]_RXP<br>PCIE[20:1]_RXN | I | Primary | Internal Pull-down | Internal Pull-down | Internal Pull-down | Off |
| PCIE_RCOMPP<br>PCIE_RCOMPN | I | Primary | Un-driven | Un-driven | Un-driven | Off |

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

## 26.5 PCI Express* Port Support Feature Details

### Table 26-1. PCI Express* Port Feature Details

| PCH | Max. Device (Ports) | Max. Lanes | PCIe* Gen Type | Encoding | Transfer Rate (MT/s) | Theoretical Max. Bandwidth (GB/s) | | |
|---|---|---|---|---|---|---|---|---|
| | | | | | | x1 | x2 | x4 |
| PCH-H | 16 | 20 | 1 | 8b/10b | 2500 | 0.25 | 0.50 | 1.00 |
| | | | 2 | 8b/10b | 5000 | 0.50 | 1.00 | 2.00 |
| | | | 3 | 128b/130b | 8000 | 1.00 | 2.00 | 3.94 |

***Notes:***
1. Theoretical Maximum Bandwidth (GB/s) = ((Transfer Rate * Encoding * # PCIe Lanes) /8)/1000
   — Gen3 Example: = ((8000 * 128/130* 4)/8)/1000 = 3.94 GB/s
2. When GbE is enabled on a PCIe Root Port, the Max. Device (Ports) value listed is reduced by a factor of 1
3. See PCH PCIe* SkU specific feature breakdown details (Max. device support, Max. lane support, PCIe* Gen type) covered within the "Introduction" chapter

### Table 26-2. PCI Express* Link Configurations Supported

| PCIe* Link Config | Flexible HSIO Lanes | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| | PCIe Controller 1 | | | | PCIe Controller 2 | | | | PCIe Controller 3 | | | | PCIe Controller 4 | | | | PCIe Controller 5 | | | |
| | PCI Express* Lanes | | | | | | | | | | | | | | | | | | | |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 1x4 | P1 | | | | P5 | | | | P9 | | | | P13 | | | | P17 | | | |
| 2x2 | P1 | | P3 | | P5 | | P7 | | P9 | | P11 | | P13 | | P15 | | P17 | | P19 | |
| 1x2 + 2x1 | P1 | | P3 | P4 | P5 | | P7 | P8 | P9 | | P11 | P12 | P13 | | P15 | P16 | P17 | | P19 | P20 |
| 2x1 + 1x2 | P4 | P3 | P1 | | P8 | P7 | P5 | | P12 | P11 | P9 | | P16 | P15 | P13 | | P20 | P19 | P17 | |
| 4x1 | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 | P13 | P14 | P15 | P16 | P17 | P18 | P19 | P20 |

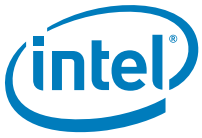
***Notes:***

1. P# refers to a specific PCH PCI Express* Root Port #; for example P3 = PCH PCI Express* Root Port 3
2. A PCIe* Lane is composed of a single pair of Transmit (TX) and Receive (RX) differential pairs, for a total of four data wires per PCIe* Lane (such as, PCIE[3]_TXP/ PCIE[3]_TXN and PCIE[3]_RXP/ PCIE[3]_RXN make up PCIe* Lane 3). A connection between two PCIe* devices is known as a PCIe* Link, and is built up from a collection of one or more PCIe* Lanes which make up the width of the link (such as bundling 2 PCIe* Lanes together would make a x2 PCIe* Link). A PCIe* Link is addressed by the lowest number PCIe* Port it connects to in the PCH (such as a x2 PCIe* Link connected to PCIe* Ports 3 and 4 would be called x2 PCIe* Port 3). This lowest number PCIe* Port in the PCIe* Link is known as the PCIe* Root Port.
3. The PCIe* Ports can be configured independently from one another but the max number of configured Devices (Ports) must not be exceeded
4. Unidentified Ports within a PCIe* Link Configuration are disabled but their physical lanes are used for the identified Port.
5. GbE can be mapped to PCIe* Root Ports 4, 5, 9, 12, or 13 within their respective PCIe* Link configurations. When GbE is enabled on a PCIe* Root Port, there can be at most up to a max of 15 Device (Ports) enabled
6. PCH-H supports up to Three x4 or x2 re-mapped (Intel® Rapid Storage Technology) PCIe* SSD Gen 1/Gen 2/Gen 3 devices with a max of One re-mapped x4 or x2 device on PCIe* Controller #3, a max of One re-mapped x4 or x2 device on PCIe* Controller #4, and a max of One re-mapped x4 or x2 device on PCIe* Controller #5
7. The SRCCLKREQ#[15:0] signals can be configured to map to any of the PCH-H PCI Express* Root Ports
8. SRCCLKREQ#[15:0] to CLKOUT_PCIE_P/N[15:0] Mapping Requirements
   - SRCCLKREQ#[7:0] signals can be mapped to any of the CLKOUT_PCIE_P/N[7:0] differential clock pairs
   - SRCCLKREQ#[15:8] signals can be mapped to any of the CLKOUT_PCIE_P/N[15:8] differential clock pairs
9. Reference and understand the PCIe* High Speed I/O Muxing details covered in the "Flexible I/O" Chapter
10. Supported Motherboard PCIe* Link Configuration Details
    - With PCIe* Controller Lane Reversal Disabled = 1x4, 2x2, 1x2+2x1, and 4x1
    - With PCIe* Controller Lane Reversal Enabled = 1x4 and 2x1+1x2
11. See PCH PCIe* Sku specific feature breakdown details (PCIe* Lane Mapping/Usage and Total Intel® RST for PCIe* Storage Devices) covered within the "Introduction" chapter

## 26.5.1 Intel® Rapid Storage Technology (Intel® RST) for PCIe* Storage

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

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



#### 26.5.1.1 Supported Features Summary

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

### 26.5.2 Interrupt Generation

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

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

Table 26-3 summarizes interrupt behavior for MSI and wire-modes. In the table "bits" refers to the hot-plug and PME interrupt bits.

**Table 26-3. MSI Versus PCI IRQ Actions**

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

### 26.5.3 Power Management

#### 26.5.3.1 S3/S4/S5 Support

Software initiates the transition to S3/S4/S5 by performing an I/O write to the Power Management Control register in the SoC. After the I/O write completion has been returned to the processor, the Power Management Controller will signal each root port to send a PME_Turn_Off message on the downstream link. The device attached to the link will eventually respond with a PME_TO_Ack followed by sending a PM_Enter_L23 DLLP (Data Link Layer Packet) request to enter L23. The Express ports and Power Management Controller take no action upon receiving a PME_TO_Ack. When all the Express port links are in state L23, the Power Management Controller will proceed with the entry into S3/S3/S5.

Prior to entering S3, software is required to put each device into $D3_{HOT}$. When a device is put into $D3_{HOT}$, it will initiate entry into a L1 link state by sending a PM_Enter_L1 DLLP. Under normal operating conditions when the root port 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.

#### 26.5.3.2 Resuming from Suspended State

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

#### 26.5.3.3 Device Initiated PM_PME Message

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

If this is the first message received (RSTS.PS), the root port will set RSTS.PS, and log the PME Requester ID into RSTS.RID. If an interrupt is enabled using RCTL.PIE, an interrupt will be generated. This interrupt can be either a pin or an MSI if MSI is enabled using MC.MSIE. See Section 26.5.3.4 for SMI/SCI generation.

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

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

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

#### 26.5.3.4 SMI/SCI Generation

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

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

When operating at PCIe* 8Gb/s, Link Equalization Request can also be routed to generate SCI or SMI. The intention8 for the SCI/SMI is to invoke the proprietary software to diagnose the reason behind the Link Equalization Request interrupt and

take the proper link recovery path, which may include software re-performing link equalization. Root Ports do not support the hardware mechanism to service the Link Equalization Request from the device.

#### 26.5.3.5 Latency Tolerance Reporting (LTR)

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

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

### 26.5.4 Dynamic Link Throttling

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

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

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

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

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

The period of the Throttling Window is configurable through the TNPT.TP field. The Throttling Window is always referenced from the time a new Throttle State change indication is received by the Root Port or from the time the throttling is enabled by the configuration register. The Throttling Window and Throttling Zone timers continue to behave the same as in L0 or L0s even if the link transitions to other LTSSM states,

except for L1, L23_Rdy and link down. For L1 case, the timer is allowed to be stopped and hardware is allowed to re-start the Throttling Window and the corresponding Throttling Zone timers on exit from L1.

### 26.5.5 Port 8xh Decode

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

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

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

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

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

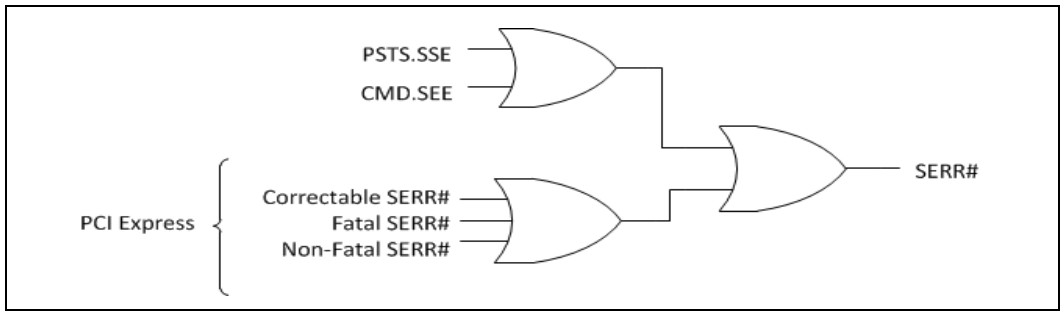
Soft straps are needed as a method to configure the port statically to operate in this mode. This mode is only enabled if the SSD connector is present on the motherboard, where the SSD connector does not include the reference clock. No change is being made to PCIe* add-in card form factors and solutions.

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

### 26.5.7 SERR# Generation

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

**Figure 26-1. Generation of SERR# to Platform**




## 26.5.8 Hot-Plug

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

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

### 26.5.8.1 Presence Detection

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

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

### 26.5.8.2 SMI/SCI Generation

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

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

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

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

## 26.5.9 PCI Express* Lane Polarity Inversion

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

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

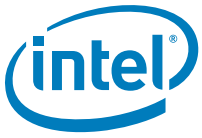
### 26.5.10 PCI Express* Controller Lane Reversal

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

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

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

§ §



# 27 Power Management

## 27.1 Acronyms

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

## 27.2 References

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

## 27.3 Overview

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

## 27.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **ACPRESENT/**GPD1 | I | **ACPRESENT:** This input pin indicates when the platform is plugged into AC power or not. In addition to the previous Intel ME to EC communication, the PCH uses this information to implement the Deep Sx policies. For example, the platform may be configured to enter Deep Sx when in S4 or S5 and only when running on battery. This is powered by Deep Sx Well. |
| **BATLOW#/**GPD0 | I | **Battery Low:** This signal is available in Mobile package only. An input from the battery to indicate that there is insufficient power to boot the system. Assertion will prevent wake from S3–S5 state. This signal can also be enabled to cause an SMI# when asserted. For Mobile package, this signal is multiplexed with GPD_0. This signal must be tied high to the VCCDSW_3p3, which will be tied to VCCPRIM_3p3 on Deep Sx disabled platforms.<br>***Note:*** Require external Pull-up to VCCDSW_3p3. |
| **BM_BUSY#** / GPP_A12 /ISH_GP6/ SX_EXIT_HOLDOFF# | I | **Bus Master Busy:** Generic bus master activity indication driven into the PCH. Can be configured to set the PM1_STS.BM_STS bit. Can also be configured to assert indications transmitted from the PCH to the processor using the PMSYNCH pin. |
| **DRAM_RESET#** | OD O | **System Memory DRAM Reset**: Active low reset signal to DRAM.<br>***Note:*** An external Pull-up to the DRAM power plane is required. |

| Name | Type | Description |
|---|---|---|
| **DSW_PWROK** | I | **DSW PWROK**: Power OK Indication for the VCCDSW_3p3 voltage rail. This input is tied together with RSMRST# on platforms that do not support Deep Sx.<br>***Note:*** This signal is in the RTC well. |
| **LAN_WAKE#**/GPD2 | I | **LAN WAKE**: is an active low wake indicator from the GbE PHY.<br>***Note:*** External Pull-up required. |
| **LANPHYPC** /GPD11 | O | **LAN PHY Power Control:** LANPHYPC is used to indicate that power needs to be restored to the Platform LAN Connect Device, when implementing Intel Auto Detect Battery Saver feature. |
| **PCH_PWROK** | I | **PCH Power OK:** When asserted, PCH_PWROK is an indication to the PCH that all of its core power rails have been stable for at least 5 ms. PCH_PWROK can be driven asynchronously. When PCH_PWROK is negated, the PCH asserts PLTRST#.<br>***Note:*** PCH_PWROK must not glitch, even if RSMRST# is low. |
| **PLTRST#**/GPP_B13 | O | **Platform Reset:** The PCH asserts PLTRST# to reset devices on the platform (such as SIO, LAN, processor, and so forth.). The PCH asserts PLTRST# during power-up and when S/W initiates a hard reset sequence through the Reset Control register (I/O port CF9h). The PCH drives PLTRST# active a minimum of 1 ms when initiated through the Reset Control register (I/O port CF9h).<br>***Note:*** PCI/PCIe* specification requires that the power rails associated with PCI/PCIe* (typically the 3.3V, 5V, and 12V core well rails) have been valid for 100 ms prior to PLTRST# de-assertion. System designers must ensure the requirement is met on the platform. |
| **PME#**/GPP_A11 | I/OD | **Power Management Event**: Driven by devices to wake the system or issue SCI. |
| **PWRBTN#**/GPD3 | I | **Power Button:** The Power Button will cause SMI# or SCI to indicate a system request to go to a sleep state. If the system is already in a sleep state, this signal will cause a wake event. If PWRBTN# is pressed for more than 4 seconds, this will cause an unconditional transition (power button override) to the S5 state. Override will occur even if the system is in the S3-S4 states. This signal has an internal Pull-up resistor and has an internal 16 ms de-bounce on the input.<br>***Note:*** Upon entry to S5 due to a power button override, if Deep Sx is enabled and conditions are met, the system will transition to Deep Sx. |
| **RSMRST#** | I | **Resume Well Reset:** This signal is used for resetting the resume power plane logic. This signal must be asserted for at least t201 after the suspend power wells are valid. When de-asserted, this signal is an indication that the suspend power wells are stable. |
| **SLP_A#/**GPD6 | O | **SLP_A#:** Used to control power to the active sleep well (ASW) of the Platform.<br>***Note:*** There is no corresponding APWROK signal input to the PCH, but the PCH does have an internally generated version of APWROK that is timed from SLP_A#. |
| **SLP_LAN#** | O | **LAN Sub-System Sleep Control:** When SLP_LAN# is de-asserted it indicates that the PHY device must be powered. When SLP_LAN# is asserted, power can be shut off to the PHY device. SLP_LAN# will always be de-asserted in S0 and anytime SLP_A# is de-asserted. |
| **SLP_WLAN#** / GPD9 | O | **WLAN Sub-System Sleep Control:** When SLP_WLAN# is asserted, power can be shut off to the external wireless LAN device. SLP_WLAN will always will be de-asserted in S0.<br>The selection between native and GPIO mode is based on a soft strap. The soft strap default is '0', slp_wlan# mode. Set soft strap to '1' to use the GPIO mode. |
| **SLP_S0#/**GPP_B12 | O | **S0 Sleep Control**: When PCH is idle and processor is in C10 state, this pin will assert to indicate VR controller can go into a light load mode. This signal can also be connected to EC for other power management related optimizations. |

| Name | Type | Description |
|---|---|---|
| **SLP_S3#**/GPD4 | O | **S3 Sleep Control:** SLP_S3# is for power plane control. This signal shuts off power to all non-critical systems when in S3 (Suspend To RAM), S4 (Suspend to Disk), or S5 (Soft Off) states. |
| **SLP_S4#**/GPD5 | O | **S4 Sleep Control**: SLP_S4# is for power plane control. This signal shuts power to all non-critical systems when in the S4 (Suspend to Disk) or S5 (Soft Off) state.<br>***Note:*** This pin must be used to control the DRAM power in order to use the PCH DRAM power-cycling feature. |
| **SLP_S5#/**GPD10 | O | **S5 Sleep Control:** SLP_S5# is for power plane control. This signal is used to shut power off to all non-critical systems when in the S5 (Soft Off) states. |
| **SLP_SUS#** | O | **Deep Sx Indication:** When asserted (driven low), this signal indicates PCH is in Deep Sx state where internal Sus power is shut off for enhanced power saving. When de-asserted (driven high), this signal indicates exit from Deep Sx state and Sus power can be applied to PCH. If Deep Sx is not supported, then this pin can be left unconnected.<br>***Note:*** This pin is in the DSW power well. |
| **SUSACK#/**GPP_A15 | I | **SUSACK#:** If Deep Sx is supported, the EC/motherboard controlling logic must change SUSACK# to match SUSWARN# once the EC/motherboard controlling logic has completed the preparations discussed in the description for the SUSWARN# pin.<br>***Note:*** SUSACK# is only required to change in response to SUSWARN# if Deep Sx is supported by the platform. |
| **SUSCLK/**GPD8 | O | **Suspend Clock:** This clock is a digitally buffer version of the RTC clock. |
| **SUSWARN#**/ SUSPWRDNACK/ GPP_A13 | O | **SUSWARN#:** This pin asserts low when the PCH is planning to enter the Deep Sx power state and remove Primary power (using SLP_SUS#). The EC/motherboard controlling logic must observe edges on this pin, preparing for SUS well power loss on a falling edge and preparing for Primary well related activity (host/Intel ME wakes and runtime events) on a rising edge. SUSACK# must be driven to match SUSWARN# once the above preparation is complete. SUSACK# should be asserted within a minimal amount of time from SUSWARN# assertion as no wake events are supported if SUSWARN# is asserted but SUSACK# is not asserted. Platforms supporting Deep Sx, but not wishing to participate in the handshake during wake and Deep Sx entry may tie SUSACK# to SUSWARN#.<br>This pin is multiplexed with SUSPWRDNACK since it is not needed in Deep Sx supported platforms. |
| **SUSPWRDNACK/** SUSWARN#/GPP_A13 | O | **SUSPWRDNACK:** Active high. Asserted by the PCH on behalf of the Intel ME when it does not require the PCH Primary well to be powered.<br>Platforms are not expected to use this signal when the PCH Deep Sx feature is used. |
| **SX_EXIT_HOLDOFF #/**GPP_A12 / BM_BUSY#/ISH_GP6 | I | **Sx Exit Holdoff Delay**: Delay exit from Sx state after SLP_A# is de-asserted. See Section 27.7.8.5 for more details. |
| **SYS_PWROK** | I | **System Power OK:** This generic power good input to the PCH is driven and utilized in a platform-specific manner. While PCH_PWROK always indicates that the core wells of the PCH are stable, SYS_PWROK is used to inform the PCH that power is stable to some other system component(s) and the system is ready to start the exit from reset. |
| **SYS_RESET#** | I | **System Reset**: This pin forces an internal reset after being de-bounced. The PCH will reset immediately if the SMBus is idle; otherwise, it will wait up to 25 ms ±2 ms for the SMBus to idle before forcing a reset on the system. |
| **VRALERT#**/GPP_B2 | I | **VR Alert:** ICC Max. throttling indicator for the PCH voltage regulators. |
| **WAKE#** | I/OD | **PCI Express* Wake Event in Sx:**<br>Input Pin in Sx. Sideband wake signal on PCI Express* asserted by components requesting wake up.<br>***Note:*** External Pull-up required. |
| **CLKRUN#**/GPP_A8 | I/OD | **LPC Clock Run:** Used to control CLKOUT_LPC[1:0]. Connects to peripherals that need to request clock restart or prevention of clock stopping. |

| Name | Type | Description |
|---|---|---|
| **SUS_STAT#**/ ESPI_RESET#/ GPP_A14 | O | **LPC Mode - Suspend Status:** This signal is asserted by the PCH to indicate that the system will be entering a low power state soon. This can be monitored by devices with memory that need to switch from normal refresh to suspend refresh mode. It can also be used by other peripherals as an indication that they should isolate their outputs that may be going to powered-off planes.<br>***Note:*** In eSPI Mode, this signal functions as ESPI Reset#. Reset signal from PCH to eSPI slave. |

## 27.5 Integrated Pull-Ups and Pull-Downs

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

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

## 27.6 I/O Signal Planes and States

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| **BATLOW#** | DSW | Undriven | Undriven | Undriven | Undriven |
| **BMBUSY#**[15] | Primary | Undriven | Undriven | Undriven | Off |
| **RSMRST#** | RTC | Undriven | Undriven | Undriven | Undriven |
| **PCH_PWROK** | RTC | Undriven | Undriven | Undriven | Undriven |
| **SYS_PWROK**[13] | Primary | Undriven | Undriven | Undriven | Off |
| **DSW_PWROK** | RTC | Undriven | Undriven | Undriven | Undriven |
| **DRAM_RESET#**[14] | DSW | Undriven | Undriven | Undriven | Undriven |
| **VR_ALERT#**[15] | Primary | Undriven | Undriven | Undriven | Off |
| **SLP_S0#**[1,6,17] | Primary | Driven High | Driven High | Driven High | Off |
| **SLP_S3#**[6,16] | DSW | Driven Low | Driven High | Driven Low | Driven Low |
| **SLP_S4#**[6,16] | DSW | Driven Low | Driven High | Driven High/ Driven Low[2] | Driven High/ Driven Low[9] |
| **SLP_S5#**[6,16] | DSW | Driven Low | Driven High | Driven High/ Driven Low[3] | Driven High/ Driven Low[9] |
| **SLP_LAN#**[6,14] | DSW | Driven Low | Driven Low | Driven High/ Driven Low[7] | Driven High/ Driven Low[7] |
| **SLP_WLAN#**[6,16] | DSW | Driven Low | Driven Low | Driven High/ Driven Low[7] | Driven High/ Driven Low[7] |
| **SLP_A#**[6,16] | DSW | Driven Low | Driven High | Driven High/ Driven Low[12] | Driven High/ Driven Low[12] |
| **SLP_SUS#**[6,14] | DSW | Driven Low | Driven High | Driven High | Driven Low |

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

***Notes:***
1. Driven High during S0 and driven Low during S0 CS.
2. SLP_S4# is driven high in S3, driven low in S4/S5.
3. SLP_S5# is driven high in S3/S4, driven low in S5.
4. In non-Deep Sx mode, pin is driven low.
5. Based on wake events and Intel ME state. SUSPWRDNACK is always '0' while in M0 or M3, but can be driven to '0' or '1' while in Moff state. SUSPWRDNACK is the default mode of operation. If Deep Sx is supported, then subsequent boots will default to SUSWARN#.
6. The pin requires glitch-free output sequence. The pad should only be pulled low momentarily when the corresponding buffer power supply is not stable.
7. Based on wake event and Intel ME state.
8. Pull-down is configurable and can be enabled in Deep Sx state; refer to DSX_CFG register (RCBA+3334h) for more details.
9. When platform enters Deep Sx, the SLP_S4# and SLP_S5# pin will retain the value it held prior to Deep Sx entry.
10. Internal weak pull resistor is default off but configurable (pu/pd/none) after boot.
11. NA
12. Pin state is a function of whether the platform is configured to have Intel ME on or off in Sx.
13. Output High-Z, not glitch free with ~20 kΩ Pull-down during respective power sequencing.
14. Output High-Z, glitch free with ~20 kΩ Pull-down during respective power sequencing
15. Output High-Z, not glitch free with ~20 kΩ Pull-down during respective power sequencing.
16. Output High-Z, glitch free with ~20 kΩ Pull-down during respective power sequencing.
17. Output High-Z, glitch free with ~20 kΩ Pull-up during respective power sequencing.

# 27.7 Functional Description

## 27.7.1 Features

- Support for *Advanced Configuration and Power Interface, Version 4.0a (ACPI)* providing power and thermal management
  - — ACPI 24-Bit Timer SCI and SMI# Generation
- PCI PME# signal for Wake Up from Low-Power states
- System Sleep State Control
  - — ACPI S3 state – Suspend to RAM (STR)

  - ACPI S4 state – Suspend-to-Disk (STD)
  - ACPI G2/S5 state – Soft Off (SOFF)
  - Power Failure Detection and Recovery
  - Deep Sx
- Intel Management Engine Power Management Support
  - Wake events from the Intel Management Engine (enabled from all S-States including Catastrophic S5 conditions)
- SLP_S0# signal for external platform VR power gating or EC power management handling during lower power condition

## 27.7.2 PCH and System Power States

Table 27-1 shows the power states defined for PCH-based platforms. The state names generally match the corresponding ACPI states.

**Table 27-1. General Power States for Systems Using the PCH**

| State/ Substates | Legacy Name/Description |
|---|---|
| G0/S0/C0 | **Full On**: Processor operating. Individual devices may be shut down or be placed into lower power states to save power. |
| G0/S0/Cx | **Cx State**: Cx states are processor power states within the S0 system state that provide for various levels of power savings. The processor manages c-state itself. The actual c-state is not passed to the PCH. Only c-state related messages are sent to the PCH and PCH will base its behavior on the actual data passed. |
| G1/S3 | **Suspend-To-RAM (STR):** The system context is maintained in system DRAM, but power is shut off to non-critical circuits. Memory is retained and refreshes continue. All external clocks stop except RTC. |
| G1/S4 | **Suspend-To-Disk (STD):** The context of the system is maintained on the disk. All power is then shut off to the system except for the logic required to resume. |
| G2/S5 | **Soft Off (SOFF):** System context is not maintained. All power is shut off except for the logic required to restart. A full boot is required when waking. |
| Deep Sx | **Deep Sx:** An optional low power state where system context may or may not be maintained depending upon entry condition. All power is shut off except for minimal logic that allows exiting Deep Sx. If Deep Sx state was entered from S3 state, then the resume path will place system back into S3. If Deep Sx state was entered from S4 state, then the resume path will place system back into S4. If Deep Sx state was entered from S5 state, then the resume path will place system back into S5. |
| G3 | **Mechanical OFF (M-Off):** System context not maintained. All power is shut off except for the RTC. No "Wake" events are possible. This state occurs if the user removes the main system batteries in a mobile system, turns off a mechanical switch, or if the system power supply is at a level that is insufficient to power the "waking" logic. When system power returns, transition will depend on the state just prior to the entry to G3 and the AFTERG3_EN bit in the GEN_PMCON_3 register (D31:F0, offset A4). Refer to Table 27-7 for more details. |

Table 27-2 shows the transitions rules among the various states.

***Note:*** Transitions among the various states may appear to temporarily transition through intermediate states. For example, in going from S0 to S4, it may appear to pass through the G1/S3 state. These intermediate transitions and states are not listed in the Table 27-2.

## Table 27-2. State Transition Rules for the PCH

| Present State | Transition Trigger | Next State |
|---|---|---|
| G0/S0/C0 | • OPI Msg<br>• SLP_EN bit set<br>• Power Button Override[3,5]<br>• Mechanical Off/Power Failure | • G0/S0/Cx<br>• G1/Sx or G2/S5 state<br>• G2/S5<br>• G3 |
| G0/S0/Cx | • OPI Msg<br>• Power Button Override[3,5]<br>• Mechanical Off/Power Failure | • G0/S0/C0<br>• S5<br>• G3 |
| G1/S3 | • Any Enabled Wake Event<br>• Power Button Override[3,5]<br>• Conditions met as described in Section 27.7.7.6.1 and Section 27.7.7.6.2<br>• Mechanical Off/Power Failure | • G0/S0/C0[2]<br>• G2/S5<br>• Deep Sx<br>• G3 |
| G1/S4 | • Any Enabled Wake Event<br>• Power Button Override[3,5]<br>• Conditions met as described in Section 27.7.7.6.1 and Section 27.7.7.6.2<br>• Mechanical Off/Power Failure | • G0/S0/C0[2]<br>• G2/S5<br>• Deep Sx<br>• G3 |
| G2/S5 | • Any Enabled Wake Event<br>• Conditions met as described in Section 27.7.7.6.1 and Section 27.7.7.6.2<br>• Mechanical Off/Power Failure | • G0/S0/C0[2]<br>• Deep Sx<br>• G3 |
| G2/Deep Sx | • Any Enabled Wake Event<br>• ACPRESENT Assertion<br>• Mechanical Off/Power Failure | • G0/S0/C0[2]<br>• G1/S3, G1/S4 or G2/S5 (see Section 27.7.7.6.2)<br>• G3 |
| G3 | • Power Returns | • S0/C0 (reboot) or G2/S5[4] (stay off until power button pressed or other wake event)[1,2] |

***Notes:***
1. Some wake events can be preserved through power failure.
2. Transitions from the S3–S5 or G3 states to the S0 state are deferred until BATLOW# is inactive in mobile configurations.
3. Includes all other applicable types of events that force the host into and stay in G2/S5.
4. If the system was in G1/S4 before G3 entry, then the system will go to S0/C0 or G1/S4.
5. Upon entry to S5 due to a power button override, if Deep Sx is enabled and conditions are met per Section 27.7.7.6, the system will transition to Deep Sx.

## 27.7.3 System Power Planes

The system has several independent power planes, as described in Table 27-3.

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

**Table 27-3. System Power Plane**

| Plane | Controlled By | Description |
|---|---|---|
| Processor | SLP_S3# signal | The SLP_S3# signal can be used to cut the power to the processor completely. |
| Main (Applicable to Platform, PCH does not have a Main well) | SLP_S3# signal | When SLP_S3# goes active, power can be shut off to any circuit not required to wake the system from the S3 state. Since the S3 state requires that the memory context be preserved, power must be retained to the main memory.<br>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.<br>***Note:*** The PCH power id not controlled by the SLP_S3# signal, but instead by the SLP_SUS# signal. |
| Memory | SLP_S4# signal<br>SLP_S5# signal | When SLP_S4# goes active, power can be shut off to any circuit not required to wake the system from the S4. Since the memory context does not need to be preserved in the S4 state, the power to the memory can also be shut down.<br>When SLP_S5# goes active, power can be shut off to any circuit not required to wake the system from the S5 state. Since the memory context does not need to be preserved in the S5 state, the power to the memory can also be shut. |
| Intel® ME | SLP_A# | SLP_A# signal is asserted when the Intel ME platform goes to M-Off. Depending on the platform, this pin may be used to control power to various devices that are part of the Intel ME sub-system in the platform. |
| LAN | SLP_LAN# | This signal is asserted in Sx/M-Off when both host and Intel ME WoL are not supported. This signal can be used to control power to the Intel GbE PHY. |
| Primary/ Suspend Well | SLP_SUS# | This signal is asserted when the Primary/Suspend rails can be externally shut off for enhanced power saving. |
| DEVICE[n] | Implementation Specific | Individual subsystems may have their own power plane. For example, GPIO signals may be used to control the power to disk drives, audio amplifiers, or the display screen. |

## 27.7.4 SMI#/SCI Generation

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

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

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

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

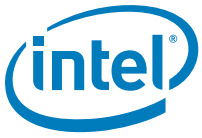


Table 27-4 shows which events can cause an SMI and SCI.

***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 27-4. Causes of SMI and SCI (Sheet 1 of 2)

| Cause | SCI | SMI | Additional Enables (Note 1) | Where Reported |
|---|---|---|---|---|
| PME# | Yes | Yes | PME_EN=1 | PME_STS |
| PME_B0 (Internal, Bus 0, PME-Capable Agents) | Yes | Yes | PME_B0_EN=1 | PME_B0_STS |
| PCI Express* PME Messages | Yes | Yes | PCI_EXP_EN=1<br>(Not enabled for SMI) | PCI_EXP_STS |
| PCI Express Hot-Plug Message | Yes | Yes | HOT_PLUG_EN=1<br>(Not enabled for SMI) | HOT_PLUG_STS |
| Power Button Press | Yes | Yes | PWRBTN_EN=1 | PWRBTN_STS |
| Power Button Override (Note 6) | Yes | No | None | PRBTNOR_STS |
| RTC Alarm | Yes | Yes | RTC_EN=1 | RTC_STS |
| ACPI Timer overflow (2.34 seconds) | Yes | Yes | TMROF_EN=1 | TMROF_STS |
| GPIO (Note 8) | Yes | Yes | | |
| LAN_WAKE# | Yes | Yes | LAN_WAKE_EN=1 | LAN_WAKE_STS |
| TCO SCI message from processor | Yes | No | None | TCOSCI_STS |
| TCO SCI Logic | Yes | No | TCOSCI_EN=1 | TCOSCI_STS |
| TCO SMI Logic | No | Yes | TCO_EN=1 | TCO_STS |
| TCO SMI – | No | Yes | None | NEWCENTURY_STS |
| TCO SMI – TCO TIMEROUT | No | Yes | None | TIMEOUT |
| TCO SMI – OS writes to TCO_DAT_IN register | No | Yes | None | SW_TCO_SMI |
| TCO SMI – Message from processor | No | Yes | None | OPISMI_STS |
| TCO SMI – NMI occurred (and NMIs mapped to SMI) | No | Yes | NMI2SMI_EN=1 | NMI2SMI_STS |
| TCO SMI – INTRUDER# signal goes active | No | Yes | INTRD_SEL=10 | INTRD_DET |
| TCO SMI – Change of the BIOSWE (D31:F0:DCh, Bit 0) bit from 0 to 1 | No | Yes | BLE=1 | BIOSWR_STS |
| TCO SMI – Write attempted to BIOS | No | Yes | BIOSWE=1 | BIOSWR_STS |
| BIOS_RLS written to 1 (Note 7) | Yes | No | GBL_EN=1 | GBL_STS |
| GBL_RLS written to | No | Yes | BIOS_EN=1 | BIOS_STS |
| Write to B2h register | No | Yes | APMC_EN = 1 | APM_STS |
| Periodic timer expires | No | Yes | PERIODIC_EN=1 | PERIODIC_STS |
| 64 ms timer expires | No | Yes | SWSMI_TMR_EN=1 | SWSMI_TMR_STS |
| Enhanced USB Legacy Support Event | No | Yes | LEGACY_USB2_EN = 1 | LEGACY_USB2_STS |
| Serial IRQ SMI reported | No | Yes | None | SERIRQ_SMI_STS |
| Device monitors match address in its range | No | Yes | None | DEVTRAP_STS |
| SMBus Host Controller | No | Yes | SMB_SMI_EN<br>Host Controller Enabled | SMBus host status reg. |
| SMBus Slave SMI message | No | Yes | None | SMBUS_SMI_STS |

**Table 27-4. Causes of SMI and SCI (Sheet 2 of 2)**

| Cause | SCI | SMI | Additional Enables (Note 1) | Where Reported |
|---|---|---|---|---|
| SMBus SMBALERT# signal active | No | Yes | None | SMBUS_SMI_STS |
| SMBus Host Notify message received | No | Yes | HOST_NOTIFY_INTREN | SMBUS_SMI_STS<br>HOST_NOTIFY_STS |
| (Mobile Only) BATLOW# assertion | Yes | Yes | BATLOW_EN=1 | BATLOW_STS |
| Access microcontroller 62h/66h | No | Yes | MCSMI_EN | MCSMI_STS |
| SLP_EN bit written to 1 | No | Yes | SLP_SMI_EN=1 | SLP_SMI_STS |
| SPI Command Completed | No | Yes | None | SPI_STS |
| eSPI SCI/SMI Request | Yes | Yes | eSPI_SCI_EN<br>See eSPI section | eSPI_SCI_STS<br>eSPI_SMI_STS |
| Software Generated GPE | Yes | Yes | SWGPE_EN=1 | SWGPE_STS |
| Intel® ME | Yes | Yes | ME_SCI_EN=1<br>ME_SCI_EN=0; ME_SMI_EN=1; | ME_SCI_STS<br>ME_SMI_STS |
| GPIO Lockdown Enable bit changes from '1' to '0' | No | Yes | GPIO_UNLOCK_SMI_EN=1 | GPIO_UNLOCK_SMI_STS |
| USB 3.0 (xHCI) SMI Event | No | Yes | XHCI_SMI_EN=1 | XHCI_SMI_STS |
| Wake Alarm Device Timer | Yes | Yes | WADT_EN | WADT_STS |

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

#### 27.7.4.1 PCI Express* SCI

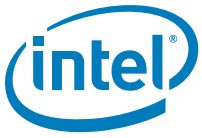
PCI Express ports and the processor have the ability to cause PME using messages. When a PME message is received, the PCH will set the PCI_EXP_STS bit. If the PCI_EXP_EN bit is also set, the PCH can cause an SCI using the GPE1_STS register.

#### 27.7.4.2 PCI Express* Hot-Plug

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

### 27.7.5 C-States

PCH-based systems implement C-states by having the processor control the states. The chipset exchanges messages with the processor as part of the C-state flow, but the chipset does not directly control any of the processors impacts of C-states, such as voltage levels or processor clocking. In addition to the messages, the PCH also provides additional information to the processor using a sideband pin (PMSYNCH).



## 27.7.6 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.

### 27.7.6.1 Conditions for Checking the 24-MHz Clock

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

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

- Cycles on LPC
- SERIRQ activity

### 27.7.6.2 Conditions for Maintaining the 24-MHz Clock

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

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

### 27.7.6.3 Conditions for Stopping the 24-MHz Clock

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

### 27.7.6.4 Conditions for Re-starting the 24-MHz Clock

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

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

## 27.7.7 Sleep States

### 27.7.7.1 Sleep State Overview

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

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

### 27.7.7.2 Initiating Sleep State

Sleep states (S3–S5) are initiated by:

- Masking interrupts, turning off all bus master enable bits, setting the desired type in the SLP_TYP field, and then setting the SLP_EN bit. The hardware then attempts to gracefully put the system into the corresponding Sleep state.
- Pressing the PWRBTN# Signal for more than 4 seconds to cause a Power Button Override event. In this case the transition to the S5 state is less graceful, since there are no dependencies on OPI messages from the processor or on clocks other than the RTC clock.
- Assertion of the THERMTRIP# signal will cause a transition to the S5 state. This can occur when system is in S0 state.
- Shutdown by integrated manageability functions (ASF/Intel AMT)
- Internal watchdog timer Timeout events

**Table 27-5. Sleep Types**

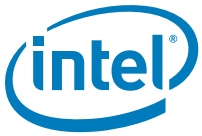
| Sleep Type | Comment |
|---|---|
| S3 | The PCH asserts SLP_S3#. The SLP_S3# signal controls the power to non-critical circuits. Power is only retained to devices needed to wake from this sleeping state, as well as to the memory. |
| S4 | The PCH asserts SLP_S3# and SLP_S4#. The SLP_S4# signal shuts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

### 27.7.7.3 Exiting Sleep States

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

Upon exit from the PCH-controlled Sleep states, the WAK_STS bit is set. The possible causes of wake events (and their restrictions) are shown in Table 27-6.

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



## Table 27-6. Causes of Wake Events

| Cause | How Enabled | Wake from Sx | Wake from Deep Sx | Wake from Sx After Power Loss (Note 2) | Wake from "Reset" Types (Note 3) |
|---|---|---|---|---|---|
| RTC Alarm | Set RTC_EN bit in PM1_EN register. | Yes | Yes | Yes | No |
| Power Button | Always enabled as Wake event. | Yes | Yes | Yes | Yes |
| Any GPIOs can be enabled for wake from the set of GPP_A to GPP_I and includes GPD[5] | | Yes | No | No | No |
| LAN_WAKE# | Enabled natively (unless pin is configured to be in GPIO mode) | Yes | Yes | Yes | Yes |
| LAN | Will use PME#. Wake enable set with LAN logic. | Yes | No | Yes | No |
| Intel® High Definition Audio | Event sets PME_B0_STS bit; PM_B0_EN must be enabled. Cannot wake from S5 state if it was entered due to power failure or power button override. | Yes | No | Yes | No |
| Primary PME# | PME_B0_EN bit in GPE0_EN[127:96] register. | Yes | No | Yes | No |
| Secondary PME# | Set PME_EN bit in GPE0_EN[127:96] register. | Yes | No | Yes | No |
| PCI Express WAKE# pin | PCIEXPWAK_DIS bit. | Yes | Yes | Yes | No |
| SMBALERT# | (Note 4) | Yes | No | Yes | Yes |
| SMBus Slave Wake Message (01h) | Wake/SMI# command always enabled as a Wake event.<br>***Note:*** SMBus Slave Message can wake the system from S3–S5, as well as from S5 due to Power Button Override. | Yes | No | Yes | Yes |
| SMBus Host Notify message received | HOST_NOTIFY_WKEN bit SMBus Slave Command register. Reported in the SMB_WAK_STS bit in the GPE0_STS register. | Yes | No | Yes | Yes |
| Intel® ME Non-Maskable Wake | Always enabled as a wake event. | Yes | No | Yes | Yes |
| Integrated WoL Enable Override | WoL Enable Override bit (in Configuration Space). | Yes | No | Yes | Yes |
| Wake Alarm Device | WADT_EN in GPE0_EN[127:96] | Yes | Yes | No | No |

***Notes:***
1. If BATLOW# signal is low, PCH will not attempt to wake from S3-S5 (nor will it exit Deep Sx), even if valid wake event occurs. This prevents the system from waking when battery power is insufficient to wake the system. However, once BATLOW# goes back high, the system will boot.
2. This column represents what the PCH would honor as wake events but there may be enabling dependencies on the device side which are not enabled after a power loss.
3. Reset Types include: Power Button override, Intel ME-initiated power button override, Intel ME-initiated host partition reset with power down, Intel ME Watchdog Timer, SMBus unconditional power down, processor thermal trip, PCH catastrophic temperature event.
4. SMBALERT# signal is multiplexed with a GPIO pin that defaults to GPIO mode. Hence, SMBALERT# related wakes are possible only when this GPIO is configured in native mode, which means that BIOS must program this GPIO to operate in native mode before this wake is possible. Because GPIO configuration is in the resume well, wakes remain possible until one of the following occurs: BIOS changes the pin to GPIO mode, a G3 occurs or Deep Sx entry occurs.
5. There are only 72 bits in the GPE registers to be assigned to GPIOs, though any of the GPIOs can trigger a wake, only those statuses of GPIO mapped to 1-tier scheme are directly accessible through the GPE status registers. For those GPIO mapped under 2-tier scheme, their status would be reflected under single master status, "GPIO_TIER2_SCI_STS" or GPE0_STS[6Fh] and further comparison needed to know which 2-tier GPI(s) has triggered the GPIO Tier 2 SCI.

### 27.7.7.4 PCI Express* WAKE# Signal and PME Event Message

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

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

### 27.7.7.5 Sx-G3-Sx, Handling Power Failures

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

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

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

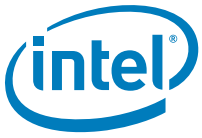
The PCH monitors both PCH_PWROK and PCH_DPWROK to detect for power failures. If PCH_PWROK goes low, the PCHPWR_FLR bit is set. If PCH_DPWROK goes low, PWR_FLR is set.

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

**Table 27-7. Transitions Due to Power Failure**

| State at Power Failure | AFTERG3_EN Bit | Transition when Power Returns |
|---|---|---|
| S0, S3 | 1<br>0 | S5<br>S0 |
| S4 | 1<br>0 | S4<br>S0 |
| S5 | 1<br>0 | S5<br>S0 |
| Deep Sx | 1<br>0 | Deep Sx[1]<br>S0 |

***Note:***
1. Entry state to Deep Sx is preserved through G3 allowing resume from Deep Sx to take appropriate path (that is, return to S3, S4 or S5).
2. Power Failure is defined as PCH_PWROK or PCH_DPWROK transition low.



### 27.7.7.6 Deep Sx

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

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

#### 27.7.7.6.1 Entry Into Deep Sx

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

All of the following must be met:

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

**Table 27-8. Supported Deep Sx Policy Configurations**

| Configuration | DPS3_EN_DC | DPS3_EN_AC | DPS4_EN_DC | DPS4_EN_AC | DPS5_EN_DC | DPS5_EN_AC |
|---|---|---|---|---|---|---|
| 1. Enabled in S5 when on Battery (ACPRESENT = 0) | 0 | 0 | 0 | 0 | 1 | 0 |
| 2. Enabled in S5 (ACPRESENT not considered) | 0 | 0 | 0 | 0 | 1 | 1 |
| 3. Enabled in S4 and S5 when on Battery (ACPRESENT = 0) | 0 | 0 | 1 | 0 | 1 | 0 |
| 4. Enabled in S4 and S5 (ACPRESENT not considered) | 0 | 0 | 1 | 1 | 1 | 1 |
| 5. Enabled in S3, S4 and S5 when on Battery (ACPRESENT = 0) | 1 | 0 | 1 | 0 | 1 | 0 |
| 6. Enabled in S3, S4 and S5 (ACPRESENT not considered) | 1 | 1 | 1 | 1 | 1 | 1 |
| 7. Deep S3/S4/ S5 disabled | 0 | 0 | 0 | 0 | 0 | 0 |
| ***Note:*** All other configurations are RESERVED. | | | | | | |

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

#### 27.7.7.6.2 Exit from Deep Sx

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

**Table 27-9. Deep Sx Wake Events**

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

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

## 27.7.8 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.

### 27.7.8.1 PWRBTN# (Power Button)

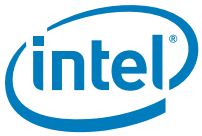
The PCH PWRBTN# signal operates as a "Fixed Power Button" as described in the *Advanced Configuration and Power Interface Specification.* PWRBTN# signal has a 16 ms de-bounce on the input. The state transition descriptions are included in Table 27-10.

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

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

**Table 27-10. Transitions Due to Power Button (Sheet 1 of 2)**

| 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<br>***Note:*** Processing of transitions starts within 100 us of the PWRBTN# input pin to PCH going low.[1] |



### Table 27-10. Transitions Due to Power Button (Sheet 2 of 2)

| Present State | Event | Transition/Action | Comment |
|---|---|---|---|
| S3 – S5 | PWRBTN# goes low | Wake Event. Transitions to S0 state | Standard wakeup<br>***Note:*** Could be impacted by SLP_* min assertion. The minimum time the PWRBTN# pin should be asserted is 150 us. The PCH will start processing this change once the minimum time requirement is satisfied.[1] |
| Deep Sx | PWRBTN# goes low | Wake Event. Transitions to S0 state | Standard wakeup<br>***Note:*** Could be impacted by SLP_* min assertion. The minimum time the PWRBTN# pin should be asserted is 150 us. The PCH will start processing this change once the minimum time requirement is satisfied but subsequently the PWRBTN# pin needs to de-assert for at least 500 us after RSMRST# de-assertion otherwise the system waits indefinitely in S5 state.[1] |
| G3 | PWRBTN# pressed | None | No effect since no power<br>Not latched nor detected<br>***Note:*** During G3 exit, PWRBTN# pin must be kept de-asserted for a minimum time of 500 us after the RSMRST# has de-asserted.[2]<br>***Note:*** Beyond this point, the minimum time the PWRBTN# pin has to be asserted to be registered by PCH as a valid wake event is 150 us.[1] |
| S0 – S4 | PWRBTN# held low for at least 4 consecutive seconds | Unconditional transition to S5 state and if Deep Sx is enabled and conditions are met per Section 27.7.7.6, the system will then transition to Deep Sx. | No dependence on processor or any other subsystem |

***Notes:***
1. If PM_CFG.PB_DB_MODE='0', the debounce logic adds 16 ms to the start/minimum time for processing of power button assertions.
2. This minimum time is independent of the PM_CFG.PB_DB_MODE value.

#### Power Button Override Function

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

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

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

**Sleep Button**

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

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

### 27.7.8.2 PME# (PCI Power Management Event)

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

There is also an internal PME_B0 bit. This is separate from the external PME# signal and can cause the same effect.

### 27.7.8.3 SYS_RESET# Signal

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

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

***Note:*** If bit 3 of the CF9h I/O register is set then SYS_RESET# will result in a full power-cycle reset.

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

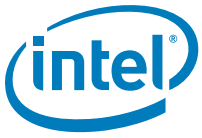
***Note:*** SYS_RESET# is in the primary power well but it only affects the system when PCH_PWROK is high.

### 27.7.8.4 THERMTRIP# Signal

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

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

If the processor is running extremely hot and is heating up, it is possible (although very unlikely) that components around it, such as the PCH, are no longer executing cycles properly. Therefore, if THERMTRIP# goes active, and the PCH is relying on state machine logic to perform the power down, the state machine may not be working, and the system will not power down.



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

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

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

#### 27.7.8.5 Sx_Exit_Holdoff#

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

Requirements to support Sx_Exit_Holdoff#:

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

After the PCH has booted up to S0 at least once since the last G3 or DeepSx exit, the EC can begin monitoring SLP_A# and using the SX_EXIT_HOLDOFF# pin to stop the PCH from accessing flash. When SLP_A# asserts, if the EC intends to access flash, it will assert SX_EXIT_HOLDOFF#. To cover the case where the PCH is going through a global reset, and not a graceful Sx+CMoff/Sx+CM3PG entry, the EC must monitor the SPI flash CS0# pin for 5ms after SLP_A# assertion before making the determination that it is safe to access flash.

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

### 27.7.9 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.

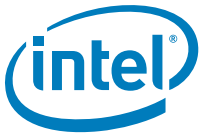
#### 27.7.9.1 Write Only Registers with Read Paths in ALT Access Mode

The registers described in Table 27-11 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 27-11. Write Only Registers with Read Paths in ALT Access Mode**

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

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



### 27.7.9.2 PIC Reserved Bits

Many bits within the PIC are reserved, and must have certain values written in order for the PIC to operate properly. Therefore, there is no need to return these values in ALT access mode. When reading PIC registers from 20h and A0h, the reserved bits shall return the values listed in Table 27-12.

**Table 27-12. PIC Reserved Bits Return Values**

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

### 27.7.9.3 Read Only Registers with Write Paths in ALT Access Mode

The registers described in Table 27-13 have write paths to them in ALT access mode. Software restores these values after returning from a powered down state. These registers must be handled special by software. When in normal mode, writing to the base address/count register also writes to the current address/count register. Therefore, the base address/count must be written first, then the part is put into ALT access mode and the current address/count register is written.

**Table 27-13. Register Write Accesses in ALT Access Mode**

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

## 27.7.10 System Power Supplies, Planes, and Signals

### 27.7.10.1 Power Plane Control

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

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

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

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

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

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

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

### 27.7.10.2 SLP_S4# and Suspend-to-RAM Sequencing

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

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

### 27.7.10.3 PCH_PWROK Signal

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

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

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

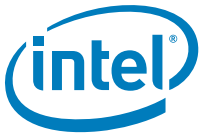
### 27.7.10.4 BATLOW# (Battery Low)

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

### 27.7.10.5 SLP_LAN# Pin Behavior

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

- The LAN PHY is always powered when the Host and Intel® ME systems are running.
  - SLP_LAN#='1' whenever SLP_S3#='1' or SLP_A#='1'.
- If the LAN PHY is required by Intel ME in Sx/M-Off or Deep Sx, Intel ME must configure SLP_LAN#='1' irrespective of the power source and the destination power state. Intel ME must be powered at least once after G3 to configure this.

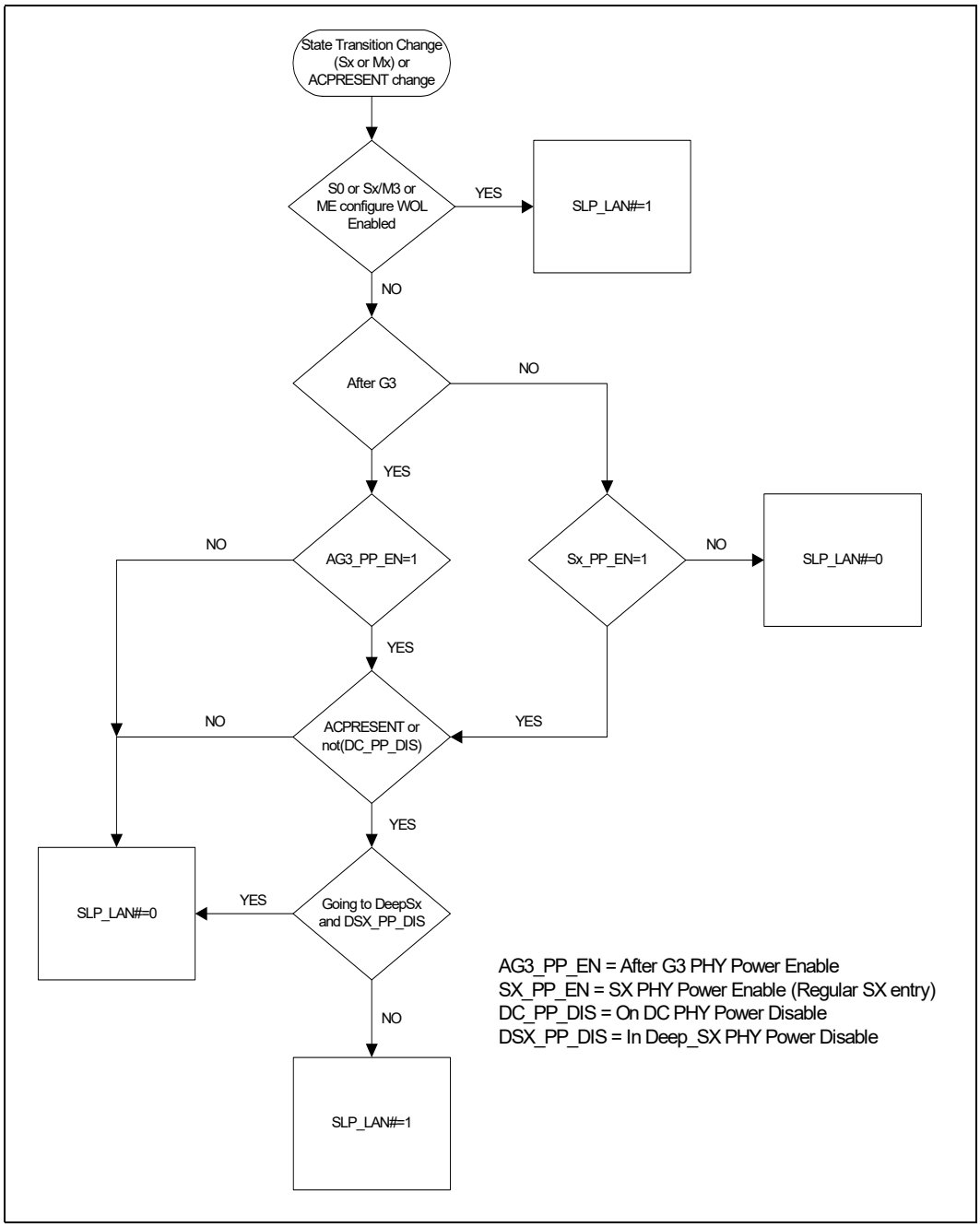


- If the LAN PHY is required after a G3 transition, the host BIOS must set AG3_PP_EN (B0:D31:F0:A0h bit 28).
- If the LAN PHY is required in Sx/M-Off, the host BIOS must set SX_PP_EN (B0:D31:F0:A0h bit 27).
- If the LAN PHY is required in Deep Sx, the host BIOS must keep DSX_PP_DIS (B0:D31:F0:A0h bit 29) cleared.
- If the LAN PHY is not required if the source of power is battery, the host BIOS must set DC_PP_DIS (B0:D31:F0:A0h bit 30).

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

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

**Figure 27-1. Conceptual Diagram of SLP_LAN#**



### 27.7.10.6 SLP_WLAN# Pin Behavior

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

- The wireless LAN PHY is always powered when the Host is running.
  - SLP_WLAN#='1' whenever SLP_S3#='1'.
- If Wake on Wireless LAN (WoWLAN) is required from S3/S4/S5 states, the host BIOS must set HOST_WLAN_PP_EN (RCBA+3318h bit 4).
- If Intel ME has access to the Wireless LAN device:
  - The Wireless LAN device must always be powered as long as Intel ME is powered. SLP_WLAN#='1' whenever SLP_A#='1'.
  - If Wake on Wireless LAN (WoWLAN) is required from M-Off state, Intel ME will configure SLP_WLAN#='1' in Sx/M-Off.

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

### 27.7.10.7 SUSPWRDNACK/SUSWARN#/GPP_A13 Steady State Pin Behavior

Table 27-14 summarizes SUSPWRDNACK/SUSWARN#/GPP_A13 pin behavior.

**Table 27-14. SUSPWRDNACK/SUSWARN#/GPP_A13 Pin Behavior**

| Pin | Deep Sx (Supported /Not-Supported) | GPP_A13 Input/Output (Determine by GP_IO_SEL bit) | Pin Value in S0 | Pin Value in Sx/M-Off | Pin Value in Sx/M3 | Pin Value in Deep Sx |
|---|---|---|---|---|---|---|
| SUSPWRDNACK | Not Supported | Native | 0 | Depends on Intel® ME power package and power source (Note 1) | 0 | Off |
| SUSWARN# | Supported | Native | 1 | 1 (Note 2) | 1 | Off |
| GPP_A13 | Don't Care | IN | High-Z | High-Z | High-Z | Off |
| | Don't Care | OUT | Depends on GPP_A13 output data value | Depends on GPP_A13 output data value | Depends on GPP_A13 output data value | Off |

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

**Table 27-15. SUSPWRDNACK During Reset**

| Reset Type (Note) | SPDA Value |
|---|---|
| power-cycle Reset | 0 |
| Global Reset | 0 |
| Straight to S5 | PCH initially drive '0' and then drive per Intel ME power policy configuration. |

***Note:*** See Table 27-16

#### 27.7.10.8 RTCRST# and SRTCRST#

RTCRST# is used to reset PCH registers in the RTC Well to their default value. If a jumper is used on this pin, it should only be pulled low when system is in the G3 state and then replaced to the default jumper position. Upon booting, BIOS should recognize that RTCRST# was asserted and clear internal PCH registers accordingly. It is imperative that this signal not be pulled low in the S0 to S5 states.

SRTCRST# is used to reset portions of the Intel Management Engine and should not be connected to a jumper or button on the platform. The only time this signal gets asserted (driven low in combination with RTCRST#) should be when the coin cell battery is removed or not installed and the platform is in the G3 state. Pulling this signal low independently (without RTCRST# also being driven low) may cause the platform to enter an indeterminate state. Similar to RTCRST#, it is imperative that SRTCRST# not be pulled low in the S0 to S5 states.

### 27.7.11 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.

#### 27.7.11.1 Mobile APM Power Management

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

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

### 27.7.12 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 4 seconds if an acknowledge from the processor is not received.

When the PCH causes a reset by asserting PLTRST# its output signals will go to their reset states as defined in Chapter 9.

A reset in which the host platform is reset and PLTRST# is asserted is called a Host Reset or Host Partition Reset. Depending on the trigger a host reset may also result in power cycling see Table 27-16 for details. If a host reset is triggered and the PCH times out before receiving an acknowledge message from the processor a Global Reset with power-cycle will occur.

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

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

Table 27-16 shows the various reset triggers.

**Table 27-16. Causes of Host and Global Resets (Sheet 1 of 2)**

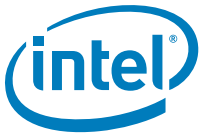
| Trigger | Host Reset Without Power Cycle[1] | Host Reset With Power Cycle[2] | Global Reset With Power Cycle[3] | Straight to S5[6] (Host Stays There) |
|---|---|---|---|---|
| Write of 0Eh to CF9h (RST_CNT Register) when CF9h when Global Reset Bit=0b | No | Yes | No (Note 4) | |
| Write of 06h to CF9h (RST_CNT Register) when CF9h when Global Reset Bit=0b | Yes | No | No (Note 4) | |
| Write of 06h or 0Eh to CF9h (RST_CNT Register) when CF9h when Global Reset Bit=1b | No | No | Yes | |
| SYS_RESET# Asserted and CF9h (RST_CNT Register) Bit 3 = 0 | Yes | No | No (Note 4) | |
| SYS_RESET# Asserted and CF9h (RST_CNT Register) Bit 3 = 1 | No | Yes | No (Note 4) | |
| SMBus Slave Message received for Reset with Power-Cycle | No | Yes | No (Note 4) | |
| SMBus Slave Message received for Reset without Power-Cycle | Yes | No | No (Note 4) | |
| SMBus Slave Message received for unconditional Power Down | No | No | No | Yes |
| TCO Watchdog Timer reaches zero two times | Yes | No | No (Note 4) | |
| Power Failure: PCH_PWROK signal goes inactive in S0 or DSW_PWROK drops | No | No | Yes | |
| SYS_PWROK Failure: SYS_PWROK signal goes inactive in S0 | No | No | Yes | |
| Processor Thermal Trip (THERMTRIP#) causes transition to S5 and reset asserts | No | No | No | Yes |
| PCH internal thermal sensors signals a catastrophic temperature condition | No | No | No | Yes |

## Table 27-16. Causes of Host and Global Resets (Sheet 2 of 2)

| Trigger | Host Reset Without Power Cycle[1] | Host Reset With Power Cycle[2] | Global Reset With Power Cycle[3] | Straight to S5[6] (Host Stays There) |
|---|---|---|---|---|
| Power Button 4 second override causes transition to S5 and reset asserts | No | No | No | Yes |
| Special shutdown cycle from processor causes CF9h-like PLTRST# and CF9h Global Reset Bit = 1 | No | No | Yes | |
| Special shutdown cycle from processor causes CF9h-like PLTRST# and CF9h Global Reset Bit = 0 and CF9h (RST_CNT Register) Bit 3 = 1 | No | Yes | No (Note 4) | |
| Special shutdown cycle from processor causes CF9h-like PLTRST# and CF9h Global Reset Bit = 0 and CF9h (RST_CNT Register) Bit 3 = 0 | Yes | No | No (Note 4) | |
| Intel® Management Engine Triggered Host Reset without Power-Cycle | Yes | No | No (Note 4) | |
| Intel® Management Engine Triggered Host Reset with Power-Cycle | No | Yes | No (Note 4) | |
| Intel® Management Engine Triggered Power Button Override | No | No | No | Yes |
| Intel® Management Engine Watchdog Timer Timeout | No | No | No | Yes |
| Intel® Management Engine Triggered Global Reset | No | No | Yes | |
| Intel® Management Engine Triggered Host Reset with power down (host stays there) | No | Yes (Note 5) | No (Note 4) | |
| PLTRST# Entry Timeout (Note 7) | No | No | Yes | |
| PROCPWRGD Stuck Low | No | No | Yes | |
| Power Management Watchdog Timer | No | No | No | Yes |
| Intel® Management Engine Hardware Uncorrectable Error | No | No | No | Yes |

***Notes:***
1. The PCH drops this type of reset request if received while the system is in S3/S4/S5.
2. PCH does not drop this type of reset request if received while system is in a software-entered S3/S4/S5 state. However, the PCH will perform the reset without executing the RESET_WARN protocol in these states.
3. The PCH does not send warning message to processor, reset occurs without delay.
4. Trigger will result in Global Reset with Power-Cycle if the acknowledge message is not received by the PCH.
5. The PCH waits for enabled wake event to complete reset.
6. Upon entry to S5, if Deep Sx is enabled and conditions are met per Section 27.7.7.6, the system will transition to Deep Sx.
7. PLTRST# Entry Timeout is automatically initiated if the hardware detects that the PLTRST# sequence has not been completed within 4 seconds of being started.

§ §



# 28 Real Time Clock (RTC)

## 28.1 Acronyms

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

## 28.2 References

None

## 28.3 Overview

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

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

The RTC also supports a date alarm that allows for scheduling a wake up event up to 30 days in advance, rather than just 24 hours in advance.

## 28.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **RTCX1** | I | **Crystal Input 1:** This signal is connected to the 32.768-KHz crystal. If no external crystal is used, then RTCX1 can be driven with the desired clock rate. Maximum voltage allowed on this pin is 1.2V. |
| **RTCX2** | O | **Crystal Input 2:** This signal is connected to the 32.768-KHz crystal. If no external crystal is used, then RTCX2 must be left floating. |
| **RTCRST#** | I | **RTC Reset:** When asserted, this signal resets register bits in the RTC well.<br>***Notes:***<br>1. Unless CMOS is being cleared (only to be done in the G3 power state), the RTCRST# input must always be high when all other RTC power planes are on.<br>2. In the case where the RTC battery is dead or missing on the platform, the RTCRST# pin must rise before the DSW_PWROK pin. |
| **SRTCRST#** | I | **Secondary RTC Reset:** This signal resets the manageability register bits in the RTC well when the RTC battery is removed.<br>***Notes:***<br>1. The SRTCRST# input must always be high when all other RTC power planes are on.<br>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. |

## 28.5 Integrated Pull-Ups and Pull-Downs

None

## 28.6 I/O Signal Planes and States

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

## 28.7 Functional Description

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

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

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

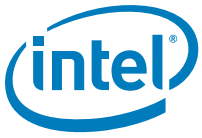
The time and calendar data should match the data mode (BCD or binary) and hour mode (12 or 24 hour) as selected in register B. It is up to the programmer to make sure that data stored in these locations is within the reasonable values ranges and represents a possible date and time. The exception to these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

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

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

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

The PCH does not implement month/year alarms.



### 28.7.1 Update Cycles

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

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

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

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

### 28.7.2 Interrupts

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

### 28.7.3 Lockable RAM Ranges

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

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

### 28.7.4 Century Rollover

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

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

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

## 28.7.5 Clearing Battery-Backed RTC RAM

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

### 28.7.5.1 Using RTCRST# to Clear CMOS

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

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

### 28.7.5.2 Using a GPI to Clear CMOS

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

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

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

## 28.7.6 External RTC Circuitry

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

**Table 28-1. RTC Crystal Requirements**

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

**Table 28-2. External Crystal Oscillator Requirements**

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

§ §

# 29 Serial ATA (SATA)

The PCH has an integrated Serial ATA (SATA) host controller with independent DMA operation on up to six ports for the PCH-H (eight ports for Server/Workstation only).

## 29.1 Acronyms

| Acronyms | Description |
|---|---|
| AHCI | Advanced Host Controller Interface |
| DMA | Direct Memory Access |
| DEVSLP | Device Sleep |
| IDE | Integrated Drive Electronics |
| RAID | Redundant Array of Independent Disks |
| SATA | Serial Advanced Technology Attachment |

## 29.2 References

| Specification | Location |
|---|---|
| Serial ATA Specification, Revision 3.2 | https://www.sata-io.org |
| Serial ATA II: Extensions to Serial ATA 1.0, Revision 1.0 | https://www.sata-io.org |
| Serial ATA II Cables and Connectors Volume 2 Gold | https://www.sata-io.org |
| Advanced Host Controller Interface Specification | http://www.intel.com/content/www/us/en/io/serial-ata/ahci.html |

## 29.3 Overview

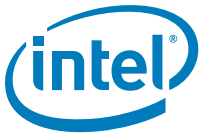
The PCH has one integrated SATA host controller that supports independent DMA operation for up to six ports for the PCH-H (eight ports for Server/Workstation only) and supports data transfer rates of up to 6 Gb/s on all ports.

The PCH SATA controller support two modes of operation, AHCI mode using memory space and RAID mode. The PCH SATA controller no longer supports IDE legacy mode using I/O space. Therefore, AHCI software is required. The PCH SATA controller supports the Serial ATA Specification, Revision 3.2.

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

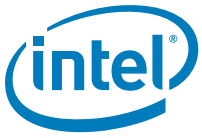
## 29.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **DEVSLP0**/ GPP_E4 | OD | **Serial ATA Port [0] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state.<br>Design Constraint: no external Pull-up or Pull-down termination required when used as DEVSLP.<br>***Note:*** This pin can be mapped to SATA Port 0. |
| **DEVSLP1**/ GPP_E5 | OD | **Serial ATA Port [1] Device Sleep:** This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state.<br>Design Constraint: no external Pull-up or Pull-down termination required when used as DEVSLP.<br>***Note:*** This pin can be mapped to SATA Port 1. |
| **DEVSLP2**/ GPP_E6 | OD | **Serial ATA Port [2] Device Sleep:** This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state.<br>Design Constraint: no external Pull-up or Pull-down termination required when used as DEVSLP.<br>***Note:*** This pin can be mapped to SATA Port 2. |
| **DEVSLP3**/ GPP_F5 | OD | **Serial ATA Port [3] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state.<br>Design Constraint: As per PDG, no external pull-up or pull-down termination required when used as DEVSLP.<br>***Note:*** This pin can be mapped to SATA Port 3. |
| **DEVSLP4**/ GPP_F6 | OD | **Serial ATA Port [4] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state.<br>Design Constraint: As per PDG, no external pull-up or pull-down termination required when used as DEVSLP.<br>***Note:*** This pin can be mapped to SATA Port 4. |
| **DEVSLP5**/ GPP_F7 | OD | **Serial ATA Port [5] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state.<br>Design Constraint: As per PDG, no external pull-up or pull-down termination required when used as DEVSLP.<br>***Note:*** This pin can be mapped to SATA Port 5. |
| **DEVSLP6**/ GPP_F8 | OD | **Serial ATA Port [6] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state.<br>Design Constraint: As per the Platform Design Guide, no external pull-up or pull-down termination required when used as DEVSLP.<br>***Note:*** This is applicable to Server/Workstation only.<br>***Note:*** This pin can be mapped to SATA Port 6. |



| Name | Type | Description |
|---|---|---|
| **DEVSLP7**/ GPP_F9 | OD | **Serial ATA Port [7] Device Sleep**: This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state.<br>Design Constraint: As per Platform Design Guide, no external pull-up or pull-down termination required when used as DEVSLP.<br>***Note:*** This is applicable to Server/Workstation only.<br>***Note:*** This pin can be mapped to SATA Port 7. |
| **SATA0A_TXP**/ PCIE9_TXP<br>**SATA0A_TXN**/ PCIE9_TXN | O | **Serial ATA Differential Transmit Pair 0 [First Instance]:** These outbound SATA Port 0 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 9 signals.<br>***Note:*** The SATA Port 0 can be configured to PCIe* Port 9 or Port 13.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 0 Strap (PCIE_SATA_P0_Flex) that select this port as SATA Port 0 or PCIe* Port 9. The default SATA/PCIe port assignment is PCIe* Port 9.<br>***Note:*** When PCIE_SATA_P0_Flex=11, the assignment of the SATA Port 0 versus PCIe* Port 9 will be based on the polarity of SATAXPCIE0. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 0 (PSCPSP_P0_STRP). |
| **SATA0A_RXP**/ PCIE9_RXP<br>**SATA0A_RXN**/ PCIE9_RXN | I | **Serial ATA Differential Receive Pair 0 [First Instance]:** These inbound SATA Port 0 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 9 signals.<br>***Note:*** The SATA Port 0 can be configured to PCIe* Port 9 or Port 13.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 0 Strap (PCIE_SATA_P0_Flex) that select this port as SATA Port 0 or PCIe* Port 9. The default SATA/PCIe port assignment is PCIe* Port 9.<br>***Note:*** When PCIE_SATA_P0_Flex=11, the assignment of the SATA Port 0 versus PCIe* Port 9 will be based on the polarity of SATAXPCIE0. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 0 (PSCPSP_P0_STRP). |
| **SATA0B_TXP**/ PCIE13_TXP<br>**SATA0B_TXN**/ PCIE13_TXN | O | **Serial ATA Differential Transmit Pair 0 [Second Instance]:** These outbound SATA Port 0 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 13 signals.<br>***Note:*** The SATA Port 0 can be configured to PCIe* Port 9 or Port 13.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 2 Strap (PCIE_SATA_P2_Flex) that select this port as SATA Port 0 or PCIe* Port 13. The default SATA/PCIe port assignment is PCIe* Port 13.<br>***Note:*** When PCIE_SATA_P2_Flex=11, the assignment of the SATA Port 0 versus PCIe* Port 13 will be based on the polarity of SATAXPCIE0. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 2 (PSCPSP_P2_STRP). |
| **SATA0B_RXP**/ PCIE13_RXP<br>**SATA0B_RXN**/ PCIE13_RXN | I | **Serial ATA Differential Receive Pair 0 [Second Instance]:** These inbound SATA Port 0 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 13 signals.<br>***Note:*** The SATA Port 0 can be configured to PCIe* Port 9 or Port 13.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 2 Strap (PCIE_SATA_P2_Flex) that select this port as SATA Port 0 or PCIe* Port 13. The default SATA/PCIe port assignment is PCIe* Port 13.<br>***Note:*** When PCIE_SATA_P2_Flex=11, the assignment of the SATA Port 0 versus PCIe* Port 13 will be based on the polarity of SATAXPCIE0. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 2 (PSCPSP_P2_STRP). |

| Name | Type | Description |
|---|---|---|
| **SATA1A_TXP**/ PCIE10_TXP **SATA1A_TXN**/ PCIE10_TXN | O | **Serial ATA Differential Transmit Pair 1 [First Instance]:** These outbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 10 signals.<br>***Note:*** The SATA Port 1 can be configured to PCIe* Port 10 or Port 14.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 1 Strap (PCIE_SATA_P1_Flex) that select this port as SATA Port 1 or PCIe* Port 10. The default SATA/PCIe port assignment is PCIe* Port 10.<br>***Note:*** When PCIE_SATA_P1_Flex=11, the assignment of the SATA Port 1 versus PCIe* Port 10 will be based on the polarity of SATAXPCIE1. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 1 (PSCPSP_P1_STRP). |
| **SATA1A_RXP**/ PCIE10_RXP **SATA1A_RXN**/ PCIE10_RXN | I | **Serial ATA Differential Receive Pair 1 [First Instance]:** These inbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 10 signals.<br>***Note:*** The SATA Port 1 can be configured to PCIe* Port 10 or Port 14.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 1 Strap (PCIE_SATA_P1_Flex) that select this port as SATA Port 1 or PCIe* Port 10. The default SATA/PCIe port assignment is PCIe* Port 10.<br>***Note:*** When PCIE_SATA_P1_Flex=11, the assignment of the SATA Port 1 versus PCIe* Port 10 will be based on the polarity of SATAXPCIE1. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 1 (PSCPSP_P1_STRP). |
| **SATA1B_TXP**/ PCIE14_TXP **SATA1B_TXN**/ PCIE14_TXN | O | **Serial ATA Differential Transmit Pair 1 [Second Instance]:** These outbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 14 signals.<br>***Note:*** The SATA Port 1 can be configured to PCIe* Port 10 or Port 14.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 3 Strap (PCIE_SATA_P3_Flex) that select this port as SATA Port 1 or PCIe* Port 14. The default SATA/PCIe port assignment is PCIe* Port 14.<br>***Note:*** When PCIE_SATA_P3_Flex=11, the assignment of the SATA Port 1 versus PCIe* Port 14 will be based on the polarity of SATAXPCIE1. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 3 (PSCPSP_P3_STRP). |
| **SATA1B_RXP**/ PCIE14_RXP **SATA1B_RXN**/ PCIE14_RXN | I | **Serial ATA Differential Receive Pair 1 [Second Instance]:** These inbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 14 signals.<br>***Note:*** The SATA Port 1 can be configured to PCIe* Port 10 or Port 14.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 3 Strap (PCIE_SATA_P3_Flex) that select this port as SATA Port 1 or PCIe* Port 14. The default SATA/PCIe port assignment is PCIe* Port 14.<br>***Note:*** When PCIE_SATA_P3_Flex=11, the assignment of the SATA Port 1 versus PCIe* Port 14 will be based on the polarity of SATAXPCIE1. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 3 (PSCPSP_P3_STRP). |
| **SATA2_TXP**/ PCIE15_TXP **SATA2_TXN**/ PCIE15_TXN | O | **Serial ATA Differential Transmit Pair 2:** These outbound SATA Port 2 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 15 signals.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 4 Strap (PCIE_SATA_P4_Flex) that select this port as SATA Port 2 or PCIe* Port 15. The default SATA/PCIe port assignment is PCIe* Port 15.<br>***Note:*** When PCIE_SATA_P4_Flex=11, the assignment of the SATA Port 2 versus PCIe* Port 15 will be based on the polarity of SATAXPCIE2. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 4 (PSCPSP_P4_STRP). |



| Name | Type | Description |
|---|---|---|
| **SATA2_RXP**/ PCIE15_RXP **SATA2_RXN**/ PCIE15_RXN | I | **Serial ATA Differential Receive Pair 2:** These inbound SATA Port 2 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 15 signals.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 4 Strap (PCIE_SATA_P4_Flex) that select this port as SATA Port 2 or PCIe* Port 15. The default SATA/PCIe port assignment is PCIe* Port 15.<br>***Note:*** When PCIE_SATA_P4_Flex=11, the assignment of the SATA Port 2 versus PCIe* Port 15 will be based on the polarity of SATAXPCIE2. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 4 (PSCPSP_P4_STRP). |
| **SATA3_TXP**/ PCIE16_TXP **SATA3_TXN**/ PCIE16_TXN | O | **Serial ATA Differential Transmit Pair 3:** These outbound SATA Port 3 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 16 signals.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 5 Strap (PCIE_SATA_P5_Flex) that select this port as SATA Port 3 or PCIe* Port 16. The default SATA/PCIe port assignment is PCIe* Port 16.<br>***Note:*** When PCIE_SATA_P5_Flex=11, the assignment of the SATA Port 3 versus PCIe* Port 16 will be based on the polarity of SATAXPCIE3. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 5 (PSCPSP_P5_STRP). |
| **SATA3_RXP**/ PCIE16_RXP **SATA3_RXN**/ PCIE16_RXN | I | **Serial ATA Differential Receive Pair 3:** These inbound SATA Port 3 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 16 signals.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 5 Strap (PCIE_SATA_P5_Flex) that select this port as SATA Port 3 or PCIe* Port 16. The default SATA/PCIe port assignment is PCIe* Port 16.<br>***Note:*** When PCIE_SATA_P5_Flex=11, the assignment of the SATA Port 3 versus PCIe* Port 16 will be based on the polarity of SATAXPCIE3. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 5 (PSCPSP_P5_STRP). |
| **SATA4_TXP**/ PCIE17_TXP **SATA4_TXN**/ PCIE17_TXN | O | **Serial ATA Differential Transmit Pair 4:** These outbound SATA Port 4 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 17 signals.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 6 Strap (PCIE_SATA_P6_Flex) that select this port as SATA Port 4 or PCIe* Port 17. The default SATA/PCIe port assignment is PCIe* Port 17.<br>***Note:*** When PCIE_SATA_P6_Flex=11, the assignment of the SATA Port 4 versus PCIe* Port 17 will be based on the polarity of SATAXPCIE4. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 6 (PSCPSP_P6_STRP). |
| **SATA4_RXP**/ PCIE17_RXP **SATA4_RXN**/ PCIE17_RXN | I | **Serial ATA Differential Receive Pair 4:** These inbound SATA Port 4 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 17 signals.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 6 Strap (PCIE_SATA_P6_Flex) that select this port as SATA Port 4 or PCIe* Port 17. The default SATA/PCIe port assignment is PCIe* Port 17.<br>***Note:*** When PCIE_SATA_P6_Flex=11, the assignment of the SATA Port 4 versus PCIe* Port 17 will be based on the polarity of SATAXPCIE4. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 6 (PSCPSP_P6_STRP). |
| **SATA5_TXP**/ PCIE18_TXP **SATA5_TXN**/ PCIE18_TXN | O | **Serial ATA Differential Transmit Pair 5:** These outbound SATA Port 5 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 18 signals.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 7 Strap (PCIE_SATA_P7_Flex) that select this port as SATA Port 5 or PCIe* Port 18. The default SATA/PCIe port assignment is PCIe* Port 18.<br>***Note:*** When PCIE_SATA_P7_Flex=11, the assignment of the SATA Port 5 versus PCIe* Port 18 will be based on the polarity of SATAXPCIE5. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 7 (PSCPSP_P7_STRP). |

| Name | Type | Description |
|---|---|---|
| **SATA5_RXP**/ PCIE18_RXP **SATA5_RXN**/ PCIE18_RXN | I | **Serial ATA Differential Receive Pair 5:** These inbound SATA Port 5 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 18 signals.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 7 Strap (PCIE_SATA_P7_Flex) that select this port as SATA Port 5 or PCIe* Port 18. The default SATA/PCIe port assignment is PCIe* Port 18.<br>***Note:*** When PCIE_SATA_P7_Flex=11, the assignment of the SATA Port 5 versus PCIe* Port 18 will be based on the polarity of SATAXPCIE5. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 7 (PSCPSP_P7_STRP). |
| **SATA6_TXP**/ PCIE19_TXP **SATA6_TXN**/ PCIE19_TXN | O | **Serial ATA Differential Transmit Pair 6:** These outbound SATA Port 6 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 19 signals.<br>***Note:*** This is applicable to Server/Workstation only.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 8 Strap (PCIE_SATA_P8_Flex) that select this port as SATA Port 6 or PCIe* Port 19. The default SATA/PCIe port assignment is PCIe* Port 19.<br>***Note:*** When PCIE_SATA_P8_Flex=11, the assignment of the SATA Port 6 versus PCIe* Port 19 will be based on the polarity of SATAXPCIE6. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 8 (PSCPSP_P8_STRP). |
| **SATA6_RXP**/ PCIE19_RXP **SATA6_RXN**/ PCIE19_RXN | I | **Serial ATA Differential Receive Pair 6:** These inbound SATA Port 6 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 19 signals.<br>***Note:*** This is applicable to Server/Workstation only.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 8 Strap (PCIE_SATA_P8_Flex) that select this port as SATA Port 6 or PCIe* Port 19. The default SATA/PCIe port assignment is PCIe* Port 19.<br>***Note:*** When PCIE_SATA_P8_Flex=11, the assignment of the SATA Port 6 versus PCIe* Port 19 will be based on the polarity of SATAXPCIE6. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 8 (PSCPSP_P8_STRP). |
| **SATA7_TXP**/ PCIE20_TXP **SATA7_TXN**/ PCIE20_TXN | O | **Serial ATA Differential Transmit Pair 7:** These outbound SATA Port 7 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 20 signals.<br>***Note:*** This is applicable to Server/Workstation only.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 9 Strap (PCIE_SATA_P9_Flex) that select this port as SATA Port 7 or PCIe* Port 20. The default SATA/PCIe port assignment is PCIe* Port 20.<br>***Note:*** When PCIE_SATA_P9_Flex=11, the assignment of the SATA Port 7 versus PCIe* Port 20 will be based on the polarity of SATAXPCIE7. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 9 (PSCPSP_P9_STRP). |
| **SATA7_RXP**/ PCIE20_RXP **SATA7_RXN**/ PCIE20_RXN | I | **Serial ATA Differential Receive Pair 7:** These inbound SATA Port 7 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.<br>The signals are multiplexed with PCIe* Port 20 signals.<br>***Note:*** This is applicable to Server/Workstation only.<br>***Note:*** Use FITC to set the soft straps of the SATA/PCIe Combo Port 9 Strap (PCIE_SATA_P9_Flex) that select this port as SATA Port 7 or PCIe* Port 20. The default SATA/PCIe port assignment is PCIe* Port 20.<br>***Note:*** When PCIE_SATA_P9_Flex=11, the assignment of the SATA Port 7 versus PCIe* Port 20 will be based on the polarity of SATAXPCIE7. Use FITC to set the soft strap of the Polarity Select SATA/PCIe Combo Port 9 (PSCPSP_P9_STRP). |
| **SATAGP0**/ SATAXPCIE0/ GPP_E0 | I | **Serial ATA Port [0] General Purpose Inputs:** When configured as SATAGP0, this is an input pin that is used as an interlock switch status indicator for SATA Port 0. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open.<br>***Note:*** The default use of this pin is GPP_E0. Pin defaults to Native mode as SATAXPCIE0 depends on soft-strap. |

| Name | Type | Description |
|---|---|---|
| **SATAGP1**/ SATAXPCIE1/ GPP_E1 | I | **Serial ATA Port [1] General Purpose Inputs:** When configured as SATAGP1, this is an input pin that is used as an interlock switch status indicator for SATA Port 1. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open.<br>***Note:*** This default use of this pin is GPP_E1. Pin defaults to Native mode as SATAXPCIE1 depends on soft-strap. |
| **SATAGP2**/ SATAXPCIE2/ GPP_E2 | I | **Serial ATA Port [2] General Purpose Inputs:** When configured as SATAGP2, this is an input pin that is used as an interlock switch status indicator for SATA Port 2. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open.<br>***Note:*** The default use of this pin is GPP_E2. Pin defaults to Native mode as SATAXPCIE2 depends on soft-strap. |
| **SATAGP3**/ SATAXPCIE3/ GPP_F0 | I | **Serial ATA Port [3] General Purpose Inputs:** When configured as SATAGP3, this is an input pin that is used as an interlock switch status indicator for SATA Port 3. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open.<br>***Note:*** The default use of this pin is GPP_F0. Pin defaults to Native mode as SATAXPCIE3 depends on soft-strap. |
| **SATAGP4**/ SATAXPCIE4/ GPP_F1 | I | **Serial ATA Port [4] General Purpose Inputs:** When configured as SATAGP4, this is an input pin that is used as an interlock switch status indicator for SATA Port 4. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open.<br>***Note:*** The default use of this pin is GPP_F1. Pin defaults to Native mode as SATAXPCIE4 depends on soft-strap. |
| **SATAGP5**/ SATAXPCIE5/ GPP_F2 | I | **Serial ATA Port [5] General Purpose Inputs:** When configured as SATAGP5, this is an input pin that is used as an interlock switch status indicator for SATA Port 5. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open.<br>***Note:*** The default use of this pin is GPP_F2. Pin defaults to Native mode as SATAXPCIE5 depends on soft-strap. |
| **SATAGP6**/ SATAXPCIE6/ GPP_F3 | I | **Serial ATA Port [6] General Purpose Inputs:** When configured as SATAGP6, this is an input pin that is used as an interlock switch status indicator for SATA Port 6. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open.<br>***Note:*** This is applicable to Server/Workstation only.<br>***Note:*** The default use of this pin is GPP_F3. Pin defaults to Native mode as SATAXPCIE6 depends on soft-strap. |
| **SATAGP7**/ SATAXPCIE7/ GPP_F4 | I | **Serial ATA Port [7] General Purpose Inputs:** When configured as SATAGP7, this is an input pin that is used as an interlock switch status indicator for SATA Port 7. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open.<br>***Note:*** This is applicable to Server/Workstation only.<br>***Note:*** The default use of this pin is GPP_F4. Pin defaults to Native mode as SATAXPCIE7 depends on soft-strap. |
| **SATALED#**/ GPP_E8 | OD O | **Serial ATA LED:** This signal is an open-drain output pin driven during SATA command activity. It is to be connected to external circuitry that can provide the current to drive a platform LED. When active, the LED is on. When tri-stated, the LED is off.<br>***Note:*** An external Pull-up resistor to VCC3_3 is required. |
| **SCLOCK**/ GPP_F10 | OD | **SGPIO Reference Clock:** 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.<br>If SGPIO interface is not used, this signal can be used as GPP_F10. |
| **SLOAD**/ GPP_F11 | OD | **SGPIO Load:** 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.<br>If SGPIO interface is not used, this signal can be used as GPP_F11. |
| **SDATAOUT0**/ GPP_F13 | OD | **SGPIO Dataout0:** 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...<br>If SGPIO interface is not used, the signals can be used as GPP_F13. |

| Name | Type | Description |
|---|---|---|
| **SDATAOUT1**/ GPP_F12 | OD | **SGPIO Dataout1:** 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...<br>If SGPIO interface is not used, the signals can be used as GPP_F12. |

## 29.5 Integrated Pull-Ups and Pull-Downs

| Signal | Resistor Type | Nominal Value | Notes |
|---|---|---|---|
| **SATAXPCIE[5:0]** | Pull-up | 20 KΩ | 1, 2 |

***Note:***
1. SATAGP[2:0]/SATAXPCIE[2:0]/GPP_E[2:0] and SATAGP[7:3]/SATAXPCIE[7:3]/GPP_F[4:0] has two native functions – the first native function (SATAXPCIEx) is selected if the Flex I/O soft strap PCIE_SATA_Px_Flex = 11b. Setting PCIE_SATA_Px_Flex = 11b also enables an internal Pull-up resistor in this pin to allow Flexible I/O selection of SATA Port x or PCIe* Port x to be assigned based on the type of card installed and based on the SATAXPCIEx mux selector with the polarity for SATA or PCIe* (When PSCPSP_Px_STRP = 0, PCIe* will be selected if the sampled value is "0" and SATA will be selected if the sampled value is "1"; When PSCPSP_Px_STRP = 1, SATA will be selected if the sampled value is "0" and PCIe* will be selected if the sampled value is "1"). Use FITC to set the soft straps of the PCIe/SATA Combo Port x Strap (PCIE_SATA_Px_Flex) and Polarity Select SATA/PCIe* Combo Port x (PSCPSP_Px_STRP). .
2. Simulation data shows that these resistor values can range from 14 KΩ – 26 KΩ.
3. SATAXPCIE[7:6] are for Server/Workstation only.

## 29.6 I/O Signal Planes and States

| Signal Name | Power Plane | During Reset | Immediately after Reset | 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[3] | Primary | Internal Pull-down | Internal Pull-down | Internal Pull-down | Off |
| **SATALED#**/GPP_E8[1] | Primary | Undriven | Undriven | Undriven | Off |
| **DEVSLP[2:0]**/ GPP_E[6:4][1] | Primary | Undriven | Undriven | Undriven | Off |
| **DEVSLP[7:3]**/ GPP_F[9:5][1,3] | Primary | Undriven | Undriven | Undriven | Off |
| **SATAGP[2:0]**/ GPP_E[2:0][2] | Primary | Undriven | Undriven | Undriven | Off |
| **SATAGP[7:3]**/ GPP_F[4:0][2,3] | Primary | Undriven | Undriven | Undriven | Off |
| **SATAXPCIE[7:0]**[2,3] | Primary | Internal Pull-up | Internal Pull-u | Undriven | Off |
| **SCLOCK**/GPP_F10[1] | Primary | Undriven | Undriven | Undriven | Off |
| **SLOAD**/GPP_F11[1] | Primary | Undriven | Undriven | Undriven | Off |
| **SDATAOUT0**/ GPP_F13[1] | Primary | Undriven | Undriven | Undriven | Off |
| **SDATAOUT1**/ GPP_F12[1] | Primary | Undriven | Undriven | Undriven | Off |

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

***Note:***
1. Pin defaults to GPIO mode. The pin state during and immediately after reset follows default GPIO mode pin state. The pin state for S0 to Deep Sx reflects assumption that GPIO Use Select register was programmed to native mode functionality. If GPIO Use Select register is programmed to GPIO mode, refer to Multiplexed GPIO (Defaults to GPIO Mode) section for the respective pin states in S0 to Deep Sx.
2. Pin defaults to Native mode as SATAXPCIEx depends on soft-strap.
3. SATA[7:6]_TXP/N, SATA[7:6]/RXP/N, DEVSLP[7:6]/GPP_F[9:8], SATAGP[7:6]/GPP_F[4:3], SATAXPCIE[7:6] are for Server/Workstation only.

# 29.7 Functional Description

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

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

The PCH SATA controller features six ports for the PCH-H (eight ports for Server/ Workstation only) that can be independently enabled or disabled (they cannot be tri-stated or driven low). Each interface is supported by an independent DMA controller.

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

## 29.7.1 SATA 6 Gb/s Support

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

## 29.7.2 SATA Feature Support

The PCH SATA controller is capable of supporting all AHCI 1.3 and AHCI 1.3.1, refer to the Intel web site on Advanced Host Controller Interface Specification for current specification status: http://www.intel.com/content/www/us/en/io/serial-ata/ahci.html.

For capability details, refer to PCH SATA controller register (D23:F0:Offset 00h CAP, and AHCI BAR PxCMD Offset 18h).

The PCH SATA controller does not support:

- Port Multiplier
- FIS Based Switching
- Command Based Switching
- IDE mode or combination mode
- Cold Presence Detect
- Function Level Reset (FLR)

### 29.7.3 Hot-Plug Operation

The PCH SATA controller supports Hot-Plug Surprise removal and Insertion Notification. An internal SATA port with a Mechanical Presence Switch can support PARTIAL and SLUMBER with Hot-Plug Enabled. Software can take advantage of power savings in the low power states while enabling Hot-Plug operation. Refer to Chapter 7 of the AHCI specification for details.

### 29.7.4 Intel® Rapid Storage Technology (Intel® RST)

The PCH SATA controller provides support for Intel® Rapid Storage Technology, providing both AHCI and integrated RAID functionality. The RAID capability provides high-performance/data-redundancy RAID 0/1/5/10 functionality on up to six ports for the PCH-H of the PCH SATA controller. Matrix RAID support is provided to allow multiple RAID levels to be combined on a single set of hard drives, such as RAID 0 and RAID 1 on two disks. Other RAID features include hot spare support, SMART alerting, and RAID 0 auto replace. Software components include an Option ROM and UEFI Driver for pre-boot configuration and boot functionality, a Microsoft* Windows* compatible driver, and a user interface for configuration and management of the RAID capability of PCH SATA controller.

***Notes:*** 1) Not all functions and capabilities may be available on all SKUs. Refer to PCH-H I/O Capabilities table and PCH-H SKUs table for details on feature availability.
2) RST only supports up to six SATA ports.

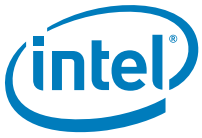
#### 29.7.4.1 Intel® Rapid Storage Technology (Intel® RST) Configuration

Intel® RST offers several diverse options for RAID (redundant array of independent disks) to meet the needs of the end user. AHCI support provides higher performance and alleviates disk bottlenecks by taking advantage of the independent DMA engines that each SATA port offers in the PCH SATA controller.

- RAID Level 0 performance scaling up to 6 drives, enabling higher throughput for data intensive applications such as video editing.
- Data redundancy is offered through RAID Level 1, which performs mirroring.
- RAID Level 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.
   Notes:



a. Not all functions and capabilities may be available on all SKUs. Refer to PCH-H I/O Capabilities table and PCH-H SKUs table for details on feature availability.
b. RST only supports up to six SATA ports.

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.

#### 29.7.4.2 Intel® Rapid Storage Technology (Intel® RST) RAID Option ROM

The Intel® Rapid Storage Technology RAID Option ROM is a standard PnP Option ROM that is easily integrated into any System BIOS. When in place, it provides the following three primary functions:

- Provides a text mode user interface that allows the user to manage the RAID configuration on the system in a pre-operating system environment. Its feature set is kept simple to keep size to a minimum, but allows the user to create and delete RAID volumes and select recovery options when problems occur.
- Provides boot support when using a RAID volume as a boot disk. It does this by providing Int13 services when a RAID volume needs to be accessed by MS-DOS applications (such as NTLDR) and by exporting the RAID volumes to the System BIOS for selection in the boot order.
- At each boot up, provides the user with a status of the RAID volumes and the option to enter the user interface by pressing CTRL-I.

### 29.7.5 Intel® Rapid Storage Technology enterprise (Intel® RSTe) - for Server/Workstation Only

The PCH SATA controller provides support for Intel® Rapid Storage Technology enterprise (RSTe), providing both AHCI and integrated RAID functionality. The RAID capability provides high-performance/data-redundancy RAID 0/1/5/10 functionality for the PCH-H 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, rebuild on hot insert, SMART alerting, TRIM and SGPIO. Software components include a legacy Option ROM and UEFI Driver for pre-boot configuration and boot functionality, a Microsoft* Windows* compatible driver, and a user interface for configuration and management of the RAID capability of PCH SATA controller.

#### 29.7.5.1 Intel® Rapid Storage Technology enterprise (Intel® RSTe) Configuration - for Server/Workstation Only

Intel® RSTe 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 8 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 enterprise, 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 enterprise functionality requires the following items:

1. PCH SKU enabled for Intel® Rapid Storage Technology enterprise.
2. Intel® Rapid Storage Technology enterprise RAID Option ROM or UEFI Driver must be on the platform.
3. Intel® Rapid Storage Technology enterprise drivers, most recent revision.
4. At least two SATA hard disk drives (minimum depends on RAID configuration).

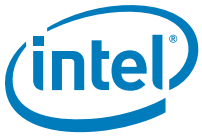
Intel® Rapid Storage Technology enterprise is only available in the following configurations:

1. The SATA controller is programmed in RAID mode, but the AIE bit (D23:F0:Offset 9Ch bit 7) is set to 0.

### 29.7.5.2 Intel® Rapid Storage Technology enterprise (Intel® RSTe) Legacy RAID Option ROM - for Server/Workstation Only

The Intel® Rapid Storage Technology enterprise Legacy 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.



#### 29.7.5.3 Intel® Rapid Storage Technology enterprise (Intel® RSTe) EFI Driver - for Server/Workstation Only

The Intel® Rapid Storage Technology enterprise EFI driver is a standard EFI driver that is easily integrated into any System BIOS. When in place, it provides the following three primary functions:

- Provides a text mode user interface (standard HII 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.

### 29.7.6 Intel® Smart Response Technology

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

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

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

### 29.7.7 Power Management Operation

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

#### 29.7.7.1 Power State Mappings

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

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

- **D0** – Device is working and instantly available.
- **D1** – Device enters when it receives a STANDBY IMMEDIATE command. Exit latency from this state is in seconds.
- **D3** – From the SATA device's perspective, no different than a D1 state, in that it is entered using the STANDBY IMMEDIATE command. However, an ACPI method is also called which will reset the device and then cut its power.

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

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

- **PHY READY** – PHY logic and PLL are both on and in active state.
- **Partial** – PHY logic is powered up, and in a reduced power state. The link PM exit latency to active state maximum is 10 ns.
- **Slumber** – PHY logic is powered up, and in a reduced power state. The link PM exit latency to active state maximum is 10 ms.
- **Devslp** – PHY logic is powered down. The link PM exit latency from this state to active state maximum is 20 ms, unless otherwise specified by DETO in Identify Device Data Log page 08h (see 13.7.9.1, 13.7.9.4 of the SATA Rev3.2 Gold specification).

Since these states have much lower exit latency than the ACPI D1 and D3 states, the SATA controller specification defines these states as sub-states of the device D0 state.

### 29.7.7.2 Power State Transitions

#### 29.7.7.2.1 Partial and Slumber State Entry/Exit

The partial and slumber states save interface power when the interface is idle. It would be most analogous to CLKRUN# (in power savings, not in mechanism), where the interface can have power saved while no commands are pending. The SATA controller defines PHY layer power management (as performed using primitives) as a driver operation from the host side, and a device proprietary mechanism on the device side. The SATA controller accepts device transition types, but does not issue any transitions as a host. All received requests from a SATA device will be ACKed.

When an operation is performed to the SATA controller such that it needs to use the SATA cable, the controller must check whether the link is in the Partial or Slumber states, and if so, must issue a COMWAKE to bring the link back online. Similarly, the SATA device must perform the same COMWAKE action.

***Note:*** SATA devices shall not attempt to wake the link using COMWAKE/COMINIT when no commands are outstanding and the interface is in Slumber.

#### 29.7.7.2.2 Devslp State Entry/Exit

Device Sleep (DEVSLP) is a host-controlled SATA interface power state. To support a hardware autonomous approach that is software agnostic Intel is recommending that BIOS configure the AHCI controller and the device to enable Device Sleep. This allows the AHCI controller and associated device to automatically enter and exit Device Sleep without the involvement of OS software.

To enter Device Sleep the link must first be in Slumber. By enabling HIPM (with Slumber) or DIPM on a Slumber capable device, the device/host link may enter the DevSleep Interface Power state.

The device must be DevSleep capable. Device Sleep is only entered when the link is in slumber, therefore when exiting the Device Sleep state, the device must resume with the COMWAKE out-of-band signal (and not the COMINIT out-of-band signal). Assuming Device Sleep was asserted when the link was in slumber, the device is expected to exit

DEVSLP to the DR_Slumber state. Devices that do not support this feature will not be able to take advantage of the hardware automated entry to Device Sleep that is part of the AHCI 1.3.1 specification and supported by Intel platforms.

#### 29.7.7.2.3 Device D1 and D3 States

These states are entered after some period of time when software has determined that no commands will be sent to this device for some time. The mechanism for putting a device in these states does not involve any work on the host controller, other than sending commands over the interface to the device. The command most likely to be used in ATA/ATAPI is the "STANDBY IMMEDIATE" command.

#### 29.7.7.2.4 Host Controller $D3_{HOT}$ State

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

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

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

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

### 29.7.7.3 Low Power Platform Consideration

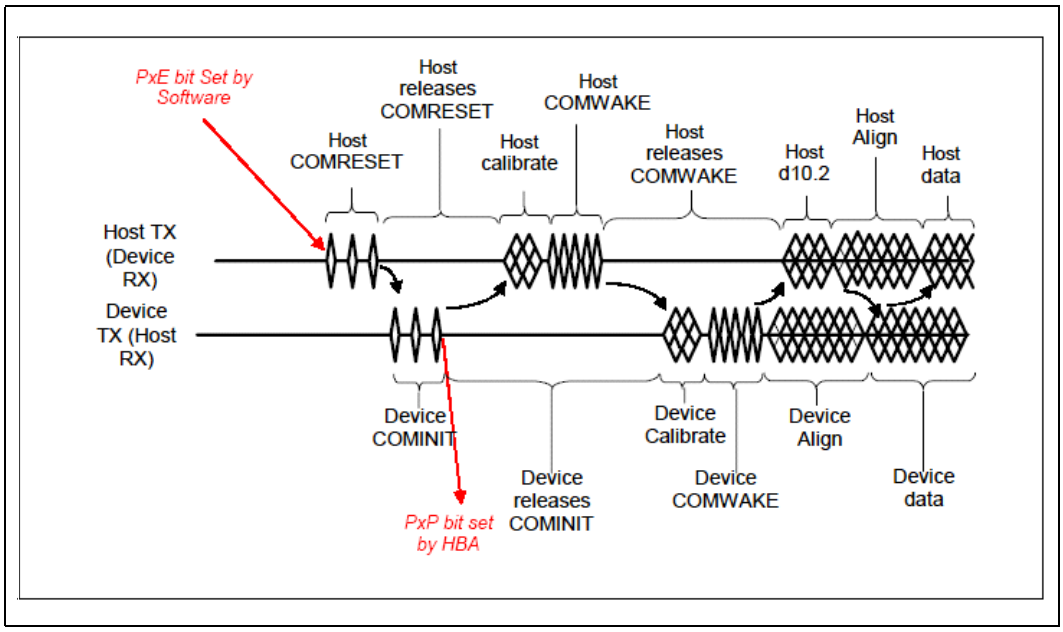
When low power feature is enabled, the Intel SATA controller may power off PLLs or OOB detection circuitry while in the Slumber link power state. As a result, a device initiated wake may not be recognized by the host. For example, when the low power feature is enabled it can prevent a Zero Power ODD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

The SATA MPHY Dynamic Power Gating (PHYDPGEPx) can be enabled/disabled for each SATA ports. Refer to SATA SIR Index 90h (for PCH-H) for the PHYDPGEPx register details.

## 29.7.8 SATA Device Presence

The flow used to indicate SATA device presence is shown in Figure 29-1. The 'PxE' 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 29-1. Flow for Port Enable/Device Present Bits**




## 29.7.9 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.

## 29.7.10 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. See Section 7.3.1 of the *AHCI Specification* for more information.

## 29.7.11 External SATA

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

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

## 29.7.12 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.

### 29.7.12.1 Mechanism

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

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

During reset all SGPIO pins will be in tri-state state. The interface will continue staying in tri-state state after reset until the first transmission occurs, when software programs the message buffer and sets the transmit bit CTL.TM. The SATA host controller will

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

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

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

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

### 29.7.12.2 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-H 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-H 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.

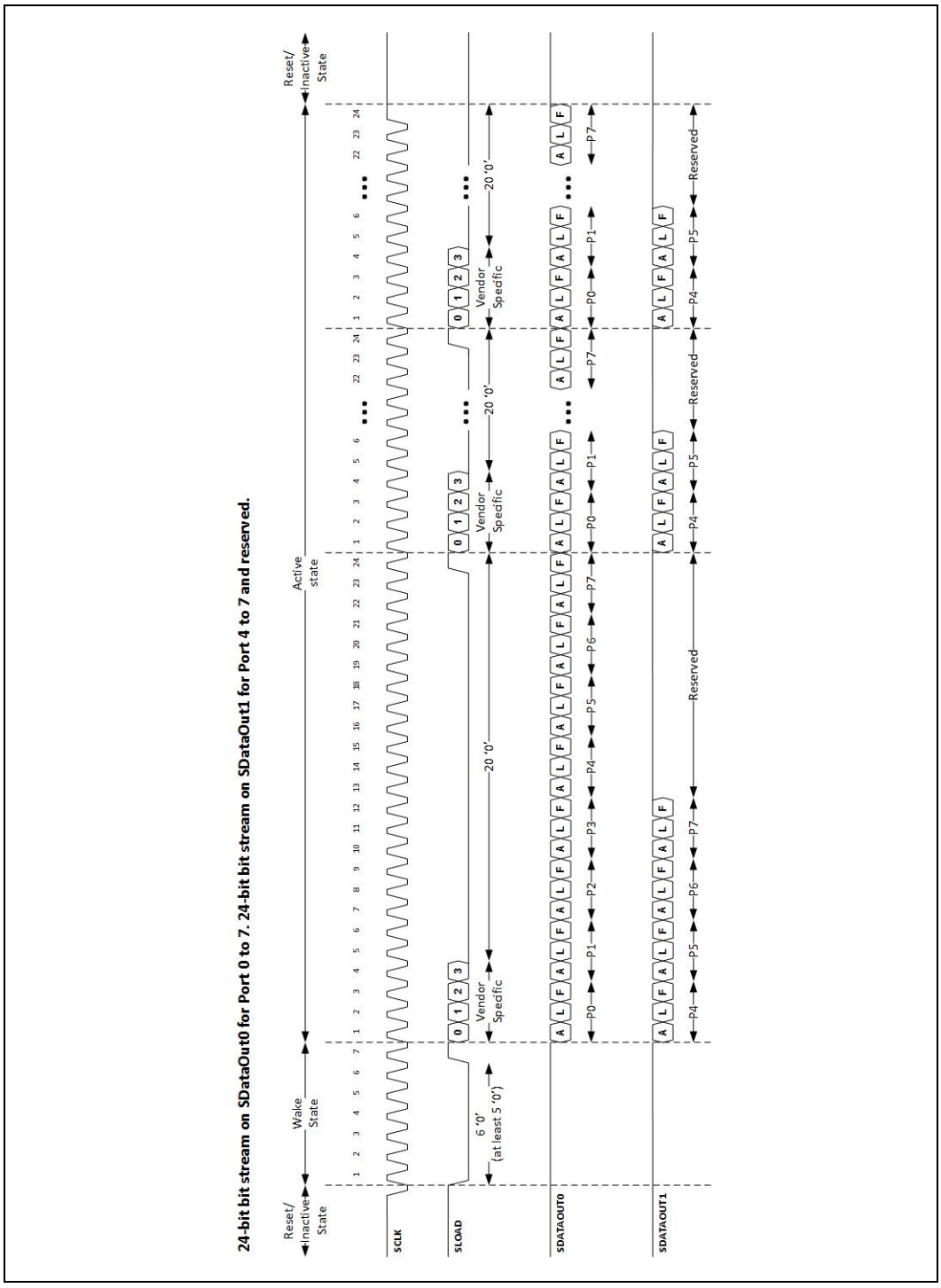
### 29.7.12.3 LED Message Type

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-H Datasheet Volume 2. The LEDs shall retain their values until there is a following update for that particular slot.

### 29.7.12.4 SGPIO Waveform

**Figure 29-2. Serial Data transmitted over the SGPIO Interface**




§ §

# 30 System Management Interface and SMLink

## 30.1 Acronyms

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

## 30.2 References

None

## 30.3 Overview

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

## 30.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **INTRUDER#** | I | **Intruder Detect:** This signal can be set to disable the system if box detected open. |
| **SML0DATA/** GPP_C4 | I/OD | **System Management Link 0 Data:** SMBus link to external PHY. External Pull-up is required. |
| **SML1CLK**/GPP_C6 | I/OD | **System Management Link 1 Clock:** SMBus link to optional Embedded Controller or BMC. External Pull-up resistor is required. |
| **SML1DATA**/ GPP_C7 | I/OD | **System Management Link 1 Data:** SMBus link to optional Embedded Controller or BMC. External Pull-up resistor is required. |
| **SML1ALERT#**/ PCHHOT#/GPP_B23 | I/OD | **System Management 1 Alert:** Alert for the Intel ME SMBus controller to optional Embedded Controller or BMC. A soft-strap determines the native function SML1ALERT# or PCHHOT# usage.<br>External Pull-up resistor is required on this pin. |
| **SML2CLK (Server Only)** /GPP_H10 | I/OD | **System Management Link 2 Clock:** External pull-up resistor is required. |
| **SML2DATA (Server Only)** / GPP_H11 | I/OD | **System Management Link 2 Data:** External pull-up resistor is required. |
| **SML2ALERT# (Server Only)** / GPP_H12 | I/OD | **System Management Link 2 Alert:** External pull-up resistor is required on this pin. |
| **SML3CLK (Server Only)** /GPP_H13 | I/OD | **System Management Link 3 Clock:** External pull-up resistor is required. |
| **SML3DATA (Server Only)** / GPP_H14 | I/OD | **System Management Link 3 Data:** External pull-up resistor is required. |

| Name | Type | Description |
|---|---|---|
| **SML3ALERT# (Server Only)** / GPP_H15 | I/OD | **System Management Link 3 Alert:** External pull-up resistor is required on this pin. |
| **SML4CLK (Server Only)** /GPP_H16 | I/OD | **System Management Link 4 Clock:** External pull-up resistor is required. |
| **SML4DATA**/ GPP_H17 | I/OD | **System Management Link 4 Data:** External pull-up resistor is required. |
| **SML4ALERT# (Server Only)** / GPP_H18 | I/OD | **System Management Link 4 Alert:** External pull-up resistor is required. |

## 30.5 Integrated Pull-Ups and Pull-Downs

| Signal | Resistor Type | Value | Notes |
|---|---|---|---|
| **SML[4:0]ALERT#** | Pull-down | 14K - 26K | The internal pull-down resistor is disable after RSMRST# de-asserted |

## 30.6 I/O Signal Planes and States

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| **INTRUDER#** | RTC | Undriven | Undriven | Undriven | Off |
| **SML[4:0]DATA** | Primary | Undriven | Undriven | Undriven | Off |
| **SML[4:0]CLK** | Primary | Undriven | Undriven | Undriven | Off |
| **SML[4:0]ALERT#** | Primary | Internal Pull-down | Driven Low | Internal Pull-down | Off |

## 30.7 Functional Description

The SMLink interfaces are controlled by the Intel® ME.

**Server Only:** Note that there are 5 SMLink interfaces in server SKUs.

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

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

Both SMLink0 and SMLink1 support up to 1 MHz.

§ §

# 31 Host System Management Bus (SMBus) Controller

## 31.1 Acronyms

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

## 31.2 References

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

## 31.3 Overview

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

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

## 31.4 Signal Description

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

## 31.5 Integrated Pull-Ups and Pull-Downs

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

## 31.6 I/O Signal Planes and States

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

## 31.7 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 $I^2C$ compatible devices.
- **Slave Interface:** Allows an external master to read from or write to the PCH. Write cycles can be used to cause certain events or pass messages, and the read cycles can be used to determine the state of various status bits. The PCH's internal host controller cannot access the PCH's internal Slave Interface.

### 31.7.1 Host Controller

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

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

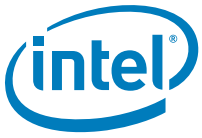
The SMBus Address Resolution Protocol (ARP) is supported by using the existing host controller commands through software, except for the Host Notify command (which is actually a received message).

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

The PCH SMBus host controller checks for parity errors as a target. If an error is detected, the detected parity error bit in the PCI Status Register is set. If bit 6 and bit 8 of the PCI Command Register are set, an SERR# is generated and the signaled SERR# bit in the PCI Status Register is set.

#### 31.7.1.1 Host Controller Operation Overview

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



The host controller supports 8 command protocols of the SMBus interface (see *System Management Bus (SMBus) Specification,* Version 2.0): Quick Command, Send Byte, Receive Byte, Write Byte/Word, Read Byte/Word, Process Call, Block Read/Write, Block Write–Block Read Process Call, and Host Notify.

The SMBus host controller requires that the various data and command fields be setup for the type of command to be sent. When software sets the START bit, the SMBus Host controller performs the requested transaction, and interrupts the processor (or generates an SMI#) when the transaction is completed. Once a START command has been issued, the values of the "active registers" (Host Control, Host Command, Transmit 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.

### 31.7.1.2 Command Protocols

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

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

#### Quick Command

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

#### Send Byte/Receive Byte

For the Send Byte command, the Transmit 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. See sections 5.5.2 and 5.5.3 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

### Write Byte/Word

The first byte of a Write Byte/Word access is the command code. The next 1 or 2 bytes are the data to be written. When programmed for a Write Byte/Word command, the Transmit 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. See section 5.5.4 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

### Read Byte/Word

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

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

### Process Call

The process call is so named because a command sends data and waits for the 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. See section 5.5.6 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

***Note:*** For process call command, the value written into bit 0 of the Transmit 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. See section 5.5.7 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

***Note:*** For Block Write, if the I2C_EN bit is set, the format of the command changes slightly. The PCH will still send the number of bytes (on writes) or receive the number of bytes (on reads) indicated in the DATA0 register. However, it will not send the contents of the DATA0 register as part of the message. Also, if the Block Write protocol sequence changes slightly, the Byte Count (bits 27:20 in the bit sequence) are not sent. As a result, the slave will not acknowledge (bit 28 in the sequence).

***Note:*** When operating in $I^2C$ mode (I2C_EN bit is set), the PCH will never use the 32-byte buffer for any block commands.

### $I^2C$* Read

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

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

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

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

The format that is used for the command is shown in Table 31-1.

**Table 31-1. $I^2C$* Block Read (Sheet 1 of 2)**

| Bit | Description |
|---|---|
| 1 | Start |
| 8:2 | Slave Address – 7 bits |
| 9 | Write |
| 10 | Acknowledge from slave |
| 18:11 | Send DATA1 register |
| 19 | Acknowledge from slave |
| 20 | Repeated Start |
| 27:21 | Slave Address – 7 bits |
| 28 | Read |

**Table 31-1. I²C* Block Read (Sheet 2 of 2)**

| Bit | Description |
|---|---|
| 29 | Acknowledge from slave |
| 37:30 | Data byte 1 from slave – 8 bits |
| 38 | Acknowledge |
| 46:39 | Data byte 2 from slave – 8 bits |
| 47 | Acknowledge |
| – | Data bytes from slave/Acknowledge |
| – | Data byte N from slave – 8 bits |
| – | NOT Acknowledge |
| – | Stop |

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

### Block Write–Block Read Process Call

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

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

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

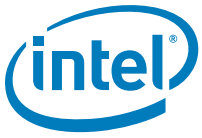
- M ≥ 1 byte
- N ≥ 1 byte
- M + N ≤ 32 bytes

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

***Note:*** There is no STOP condition before the repeated START condition, and that a NACK signifies the end of the read transfer.

***Note:*** E32B bit in the Auxiliary Control register must be set when using this protocol.

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



### 31.7.1.3 Bus Arbitration

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

If the PCH sees that it has lost arbitration, the condition is called a collision. The PCH will set the BUS_ERR bit in the Host Status Register, and if enabled, generate an interrupt or SMI#. The processor is responsible for restarting the transaction.

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

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

### 31.7.1.4 Clock Stretching

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

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

### 31.7.1.5 Bus Timeout (PCH as SMBus Master)

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

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

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

### 31.7.1.6 Interrupts/SMI#

The PCH SMBus controller uses PIRQB# as its interrupt pin. However, the system can alternatively be set up to generate SMI# instead of an interrupt, by setting the SMBUS_SMI_EN bit.

Table 31-2, Table 31-3 and Table 31-4 specify how the various enable bits in the SMBus function control the generation of the interrupt, Host and Slave SMI, and Wake internal signals. The rows in the tables are additive, which means that if more than one row is true for a particular scenario then the Results for all of the activated rows will occur.

**Table 31-2. Enable for SMBALERT#**

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

**Table 31-3. Enables for SMBus Slave Write and SMBus Host Events**

| Event | INTREN (Host Control I/O Register, Offset 02h, Bit 0) | SMB_SMI_EN (Host Configuration Register, D31: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 |

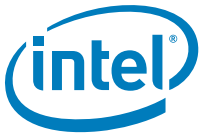
**Table 31-4. Enables for the Host Notify Command**

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

### 31.7.1.7 SMBus CRC Generation and Checking

If the AAC bit is set in the Auxiliary Control register, the PCH automatically calculates and drives CRC at the end of the transmitted packet for write cycles, and will check the CRC for read cycles. It will not transmit the contents of the PEC register for CRC. The PEC bit must not be set in the Host Control register if this bit is set, or unspecified behavior will result.

If the read cycle results in a CRC error, the DEV_ERR bit and the CRCE bit in the Auxiliary Status register at Offset 0Ch will be set.



## 31.7.2 SMBus Slave Interface

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

Key features and capabilities include:

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

***Note:*** The external microcontroller should not attempt to access the PCH SMBus slave logic until either:

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

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

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

### 31.7.2.1 Format of Slave Write Cycle

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

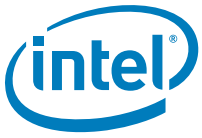
Table 31-5 has the values associated with the registers.

**Table 31-5. Slave Write Registers**

| Register | Function |
|---|---|
| 0 | Command Register. See Table 31-6 for valid values written to this register. |
| 1–3 | Reserved |
| 4 | Data Message Byte 0 |
| 5 | Data Message Byte 1 |
| 6–7 | Reserved |
| 8 | Reserved |
| 9–FFh | Reserved |
| ***Note:*** The external microcontroller is responsible to make sure that it does not update the contents of the data byte registers until they have been read by the system processor. The PCH overwrites the old value with any new value received. A race condition is possible where the new value is being written to the register just at the time it is being read. The PCH will not attempt to cover this race condition (that is, unpredictable results in this case). | |

**Table 31-6. Command Types (Sheet 1 of 2)**

| Command Type | Description |
|---|---|
| 0 | Reserved |
| 1 | **WAKE/SMI#.** This command wakes the system if it is not already awake. If system is already awake, an SMI# is generated.<br>***Note:*** The SMB_WAK_STS bit will be set by this command, even if the system is already awake. The SMI handler should then clear this bit. |
| 2 | **Unconditional Powerdown.** This command sets the PWRBTNOR_STS bit, and has the same effect as the Powerbutton Override occurring. |
| 3 | **HARD RESET WITHOUT CYCLING:** This command causes a hard reset of the system (does not include cycling of the power supply). This is equivalent to a write to the CF9h register with Bits 2:1 set to 1, but Bit 3 set to 0.<br>***Note:*** This command is only available in S0. All attempts to trigger a host reset without power cycle while the system is in Sx are dropped |
| 4 | **HARD RESET SYSTEM.** This command causes a hard reset of the system (including cycling of the power supply). This is equivalent to a write to the CF9h register with Bits 3:1 set to 1.<br>**Note:** The command is supported in the following scenarios:<br>• If the system is in Sx/M3or Sx/M3-PG, the command is supported.<br>• If the system is in Sx/Moff, the command is supported if performed after a graceful Sx entry (i.e. if the platform was put to sleep or turned off via a write to the SLP_TYP/ SLP_EN fields by the OS or BIOS). Otherwise, the command is not supported. |
| 5 | **Disable the TCO Messages.** This command will disable the PCH from sending Heartbeat and Event messages. Once this command has been executed, Heartbeat and Event message reporting can only be re-enabled by assertion and de-assertion of the RSMRST# signal. |
| 6 | **WD RELOAD:** Reload watchdog timer. |
| 7 | Reserved |



### Table 31-6. Command Types (Sheet 2 of 2)

| Command Type | Description |
|---|---|
| 8 | **SMLINK_SLV_SMI.** When the PCH detects this command type while in the S0 state, it sets the SMLINK_SLV_SMI_STS bit. This command should only be used if the system is in an S0 state. If the message is received during S3–S5 states, the PCH acknowledges it, but the SMLINK_SLV_SMI_STS bit does not get set.<br>***Note:*** It is possible that the system transitions out of the S0 state at the same time that the SMLINK_SLV_SMI command is received. In this case, the SMLINK_SLV_SMI_STS bit may get set but not serviced before the system goes to sleep. Once the system returns to S0, the SMI associated with this bit would then be generated. Software must be able to handle this scenario. |
| 9–FFh | Reserved. |

## 31.7.2.2 Format of Read Command

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

### Table 31-7. Slave Read Cycle Format

| Bit | Description | Driven By | Comment |
|---|---|---|---|
| 1 | Start | External Microcontroller | |
| 2–8 | Slave Address - 7 bits | External Microcontroller | Must match value in Receive Slave Address register |
| 9 | Write | External Microcontroller | Always 0 |
| 10 | ACK | PCH | |
| 11–18 | Command code – 8 bits | External Microcontroller | Indicates which register is being accessed. See Table 31-8 for a list of implemented registers. |
| 19 | ACK | PCH | |
| 20 | Repeated Start | External Microcontroller | |
| 21–27 | Slave Address - 7 bits | External Microcontroller | Must match value in Receive Slave Address register |
| 28 | Read | External Microcontroller | Always 1 |
| 29 | ACK | PCH | |
| 30–37 | Data Byte | PCH | Value depends on register being accessed. See Table 31-8 for a list of implemented registers. |
| 38 | NOT ACK | External Microcontroller | |
| 39 | Stop | External Microcontroller | |

### Table 31-8. Data Values for Slave Read Registers (Sheet 1 of 2)

| Register | Bits | Description |
|---|---|---|
| 0 | 7:0 | Reserved for capabilities indication. Should always return 00h. Future chips may return another value to indicate different capabilities. |
| 1 | 2:0 | **System Power State**<br>000 = S0<br>011 = S3<br>100 = S4<br>101 = S5<br>Others = Reserved |
| | 7:3 | Reserved |

## Table 31-8. Data Values for Slave Read Registers (Sheet 2 of 2)

| Register | Bits | Description |
|---|---|---|
| 2 | 3:0 | Reserved |
| | 7:4 | Reserved |
| 3 | 5:0 | **Watchdog Timer current value**<br>***Note:*** The Watchdog Timer has 10 bits, but this field is only 6 bits. If the current value is greater than 3Fh, the PCH will always report 3Fh in this field. |
| | 7:6 | Reserved |
| 4 | 0 | **Intruder Detect**. 1 = The Intruder Detect (INTRD_DET) bit is set. This indicates that the system cover has probably been opened. |
| | 1 | **Temperature Event**. 1 = Temperature Event occurred. This bit will be set if the PCH's THRM# input signal is active. Else this bit will read "0." |
| | 2 | **DOA Processor Status**. This bit will be 1 to indicate that the processor is dead |
| | 3 | 1 = **SECOND_TO_STS** bit set. This bit will be set after the second Timeout (SECOND_TO_STS bit) of the Watchdog Timer occurs. |
| | 6:4 | Reserved. Will always be 0, but software should ignore. |
| | 7 | **SMBALERT# Status.** Reflects the value of the SMBALERT# pin (when the pin is configured to SMBALERT#). Valid only if SMBALERT_DISABLE = 0. Value always returns 1 if SMBALERT_DISABLE = 1. |
| 5 | 0 | **FWH bad bit.** This bit will be 1 to indicate that the FWH read returned FFh, which indicates that it is probably blank. |
| | 1 | **Battery Low Status.** 1 if the BATLOW# pin a low. |
| | 2 | **SYS_PWROK Failure Status:** This bit will be 1 if the SYSPWR_FLR bit in the GEN_PMCON_2 register is set. |
| | 3 | Reserved |
| | 4 | Reserved |
| | 5 | **POWER_OK_BAD:** Indicates the failure core power well ramp during boot/resume. This bit will be active if the SLP_S3# pin is de-asserted and PCH_PWROK pin is not asserted. |
| | 6 | **Thermal Trip**: This bit will shadow the state of processor Thermal Trip status bit (CTS). Events on signal will not create an event message |
| | 7 | Reserved: Default value is "X"<br>***Note:*** Software should not expect a consistent value when this bit is read through SMBUS/SMLink |
| 6 | 7:0 | Contents of the Message 1 register. |
| 7 | 7:0 | Contents of the Message 2 register. |
| 8 | 7:0 | Contents of the WDSTATUS register. |
| 9 | 7:0 | Seconds of the RTC |
| A | 7:0 | Minutes of the RTC |
| B | 7:0 | Hours of the RTC |
| C | 7:0 | "Day of Week" of the RTC |
| D | 7:0 | "Day of Month" of the RTC |
| E | 7:0 | Month of the RTC |
| F | 7:0 | Year of the RTC |
| 10h–FFh | 7:0 | Reserved |

#### 31.7.2.2.1 Behavioral Notes

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

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

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

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

### 31.7.2.3 Slave Read of RTC Time Bytes

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

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

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

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

### 31.7.2.4 Format of Host Notify Command

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

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

Table 31-9 shows the Host Notify format.

**Table 31-9. Host Notify Format**

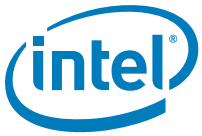
| Bit | Description | Driven By | Comment |
|---|---|---|---|
| 1 | Start | External Master | |
| 8:2 | SMB Host Address – 7 bits | External Master | Always 0001_000 |
| 9 | Write | External Master | Always 0 |
| 10 | ACK (or NACK) | PCH | PCH NACKs if HOST_NOTIFY_STS is 1 |
| 17:11 | Device Address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

### 31.7.2.5 Format of Read Command

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

**Table 31-10. 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. See Table 31-11 for a list of implemented registers. |
| 19 | ACK | PCH | |
| 20 | Repeated Start | External Microcontroller | |
| 21–27 | Slave Address - 7 bits | External Microcontroller | Must match value in Receive Slave Address register |
| 28 | Read | External Microcontroller | Always 1 |
| 29 | ACK | PCH | |
| 30–37 | Data Byte | PCH | Value depends on register being accessed. See Table 31-11 for a list of implemented registers. |
| 38 | NOT ACK | External Microcontroller | |
| 39 | Stop | External Microcontroller | |



## Table 31-11. Data Values for Slave Read Registers (Sheet 1 of 2)

| Register | Bits | Description |
|---|---|---|
| 0 | 7:0 | Reserved for capabilities indication. Should always return 00h. Future chips may return another value to indicate different capabilities. |
| 1 | 2:0 | **System Power State**<br>000 = S0<br>011 = S3<br>100 = S4<br>101 = S5<br>Others = Reserved |
| | 7:3 | Reserved |
| 2 | 3:0 | Reserved |
| | 7:4 | Reserved |
| 3 | 5:0 | **Watchdog Timer current value**<br>***Note:*** The Watchdog Timer has 10 bits, but this field is only 6 bits. If the current value is greater than 3Fh, the PCH will always report 3Fh in this field. |
| | 7:6 | Reserved |
| 4 | 0 | **Intruder Detect.** 1 = The Intruder Detect (INTRD_DET) bit is set. This indicates that the system cover has probably been opened. |
| | 1 | **Temperature Event.** 1 = Temperature Event occurred. This bit will be set if the PCH's THRM# input signal is active. Else this bit will read "0." |
| | 2 | **DOA Processor Status**. This bit will be 1 to indicate that the processor is dead |
| | 3 | 1 = **SECOND_TO_STS** bit set. This bit will be set after the second Timeout (SECOND_TO_STS bit) of the Watchdog Timer occurs. |
| | 6:4 | Reserved. Will always be 0, but software should ignore. |
| | 7 | **SMBALERT# Status.** Reflects the value of the GPIO11/SMBALERT# pin (when the pin is configured as SMBALERT#). Valid only if SMBALERT_DISABLE = 0. Value always returns 1 if SMBALERT_DISABLE = 1.<br>(high = 1, low = 0). |
| 5 | 0 | **FWH bad bit.** This bit will be 1 to indicate that the FWH read returned FFh, which indicates that it is probably blank. |
| | 1 | **Battery Low Status**. 1 if the BATLOW# pin is a 0. |
| | 2 | **SYS_PWROK Failure Status:** This bit will be 1 if the SYSPWR_FLR bit in the GEN_PMCON_2 register is set. |
| | 3 | Reserved |
| | 4 | Reserved |
| | 5 | **POWER_OK_BAD.** Indicates the failure core power well ramp during boot/resume. This bit will be active if the SLP_S3# pin is de-asserted and PCH_PWROK pin is not asserted. |
| | 6 | **Thermal Trip**. This bit will shadow the state of processor Thermal Trip status bit (CTS). Events on signal will not create an event message |
| | 7 | Reserved: Default value is "X"<br>***Note:*** Software should not expect a consistent value when this bit is read through SMBUS/SMLink |
| 6 | 7:0 | Contents of the Message 1 register. |
| 7 | 7:0 | Contents of the Message 2 register. |
| 8 | 7:0 | Contents of the WDSTATUS register. |
| 9 | 7:0 | Seconds of the RTC |
| A | 7:0 | Minutes of the RTC |
| B | 7:0 | Hours of the RTC |

### Table 31-11. Data Values for Slave Read Registers (Sheet 2 of 2)

| Register | Bits | Description |
|---|---|---|
| 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 31-12. Enables for SMBus Slave Write and SMBus Host Events

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

§ §

# 32 Serial Peripheral Interface (SPI)

## 32.1 Acronyms

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

## 32.2 References

None

## 32.3 Overview

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

The SPI interfaces support either 1.8V or 3.3V.

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

## 32.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **SPI0_CLK** | O | **SPI Clock**: SPI clock signal for the common flash/TPM interface. Supports 17 MHz, 30 MHz and 48 MHz. |
| **SPI0_CS0#** | O | **SPI Chip Select 0**: Used to select the primary SPI Flash device.<br>***Note:*** This signal cannot be used for any other type of device than SPI Flash. |
| **SPI0_CS1#** | O | **SPI Chip Select 1**: Used to select an optional secondary SPI Flash device.<br>***Note:*** This signal cannot be used for any other type of device than SPI Flash. |
| **SPI0_CS2#** | O | **SPI Chip Select 2:** Used to select the TPM device if it is connected to the SPI interface; it cannot be used for any other type of device.<br>***Note:*** TPM can be configured through soft straps to operate over LPC or SPI, but no more than 1 TPM is allowed in the system. |
| **SPI0_MOSI** | I/O | **SPI Master OUT Slave IN**: Defaults as a data output pin for PCH in Dual Output Fast Read mode. Can be configured with a Soft Strap as a bidirectional signal (SPI_IO0) to support the new Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes. |

| Name | Type | Description |
|---|---|---|
| **SPI0_MISO** | I/O | **SPI Master IN Slave OUT**: Defaults as a data input pin for PCH in Dual Output Fast Read mode. Can be configured with a Soft Strap as a bidirectional signal (SPI_IO1) to support the new Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes. |
| **SPI0_IO[3:2]** | I/O | **SPI Data I/O:** A bidirectional signal used to support Dual I/O Fast Read, Quad I/O Fast Read and Quad Output Fast Read modes. This signal is not used in Dual Output Fast Read mode. |

## 32.5 Integrated Pull-Ups and Pull-Downs

| Signal | Resistor Type | Value | Notes |
|---|---|---|---|
| **SPI0_CLK** | Pull-down | 15K - 40K | |
| **SPI0_MOSI** | Pull-up | 15K - 40K | |
| **SPI0_MISO** | Pull-up | 15K - 40K | |
| **SPI0_CS[2:0]#** | Pull-down | 15K - 40K | |
| **SPI0_IO[2:3]** | Pull-up | 15K - 40K | |

## 32.6 I/O Signal Planes and States

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

# 32.7 Functional Description

## 32.7.1 SPI for Flash

### 32.7.1.1 Overview

The PCH supports up to two SPI flash devices using two separate Chip Select pins. The maximum size of flash supported is determined by the SFDP-discovered addressing capability of each device. Each component can be up to 16 MB using 3-byte addressing or 64 MB using 4-byte addressing.

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

The SPI interface supports either 3.3V or 1.8V.

A SPI Flash device on Chip Select 0 (SPI_CS0#) with a valid descriptor MUST be attached directly to the PCH.

The PCH supports fast read which consist of:

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

The PCH SPI has a third chip select SPI_CS2# for TPM support over SPI. TPM Bus will use SPI_CLK, SPI_MISO, SPI_MOSI and SPI_CS2# SPI signals.

***Notes:***

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

### 32.7.1.2 SPI Supported Features

#### 32.7.1.2.1 Descriptor Mode

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

#### 32.7.1.2.2 SPI Flash Regions

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

**Table 32-1. SPI Flash Regions (Sheet 1 of 2)**

| Region | Content |
|---|---|
| 0 | Flash Descriptor |
| 1 | BIOS |
| 2 | Intel Management Engine |

**Table 32-1. SPI Flash Regions (Sheet 2 of 2)**

| Region | Content |
|---|---|
| 3 | Gigabit Ethernet |
| 4 | Platform Data |
| 8 | EC |

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

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

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

#### Flash Region Sizes

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

**Table 32-2. Region Size Versus Erase Granularity of Flash Components**

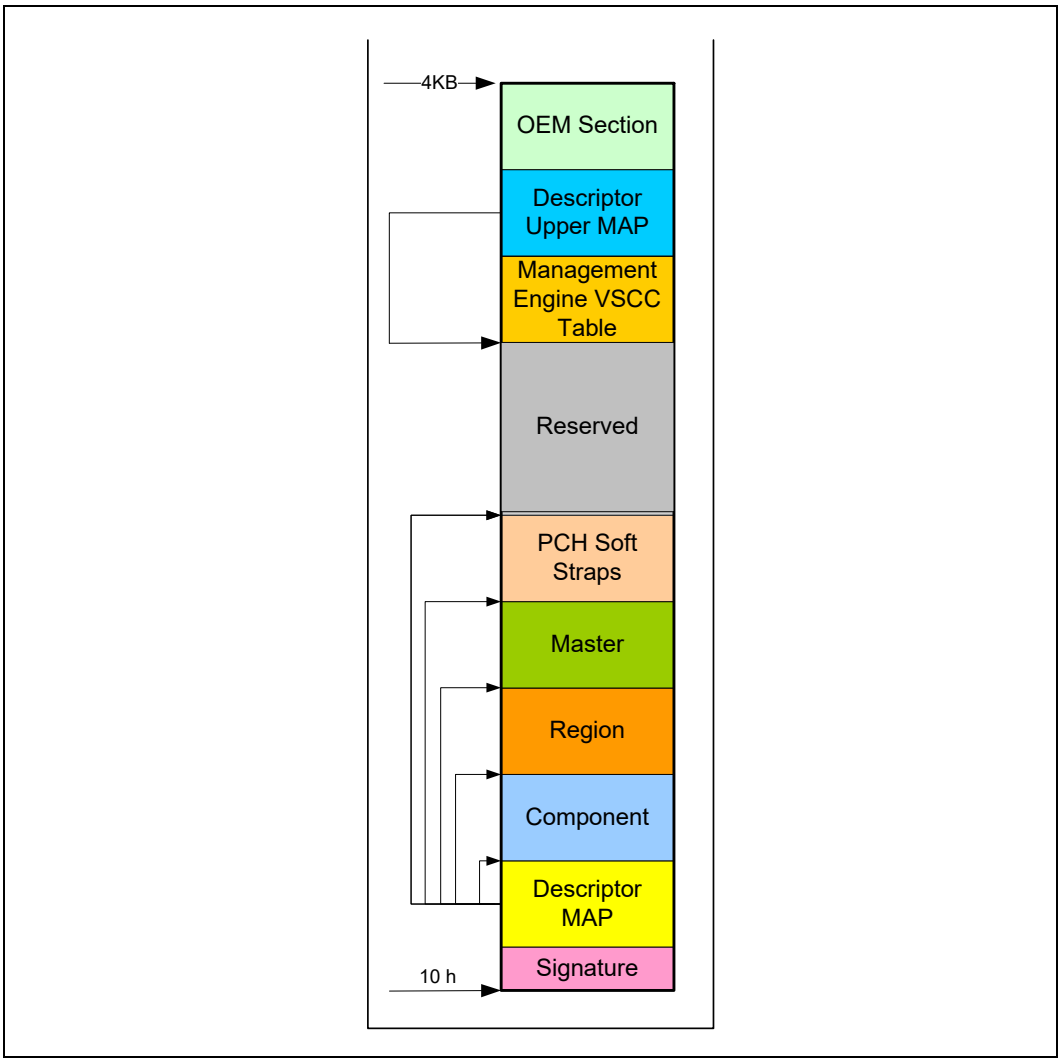
| Region | Size with 4-KB Blocks | Size with 8-KB Blocks | Size with 64-KB Blocks |
|---|---|---|---|
| Descriptor | 4 KB | 8 KB | 64 KB |
| GbE | 8 KB | 16 KB | 128 KB |
| BIOS | Varies by Platform | Varies by Platform | Varies by Platform |
| Intel® ME | Varies by Platform | Varies by Platform | Varies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

### 32.7.1.3 Flash Descriptor

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

The Flash Descriptor is made up of eleven sections as shown in Figure 32-1.

### Figure 32-1. Flash Descriptor Regions




- The Flash signature selects Descriptor Mode as well as verifies if the flash is programmed and functioning. The data at the bottom of the flash (offset 10h) must be 0FF0A55Ah in order to be in Descriptor mode.
- The Descriptor map has pointers to the other five descriptor sections as well as the size of each.
- The component section has information about the SPI flash in the system including: the number of components, density of each, invalid instructions (such as chip erase), and frequencies for read, fast read and write/erase instructions.
- The Region section points to the three other regions as well as the size of each region.
- The 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 processor and PCH Soft Strap sections contain processor and PCH configurable parameters.

- The Reserved region between the top of the processor strap section and the bottom of the OEM Section is reserved for future chipset usages.
- The Descriptor Upper MAP determines the length and base address of the Management Engine VSCC Table.
- The Management Engine VSCC Table holds the JEDEC ID and the VSCC information of the entire SPI Flash supported by the NVM image.
- OEM Section is 256 bytes reserved at the top of the Flash Descriptor for use by OEM.

#### 32.7.1.3.1 Descriptor Master Region

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

**Table 32-3. Region Access Control Table**

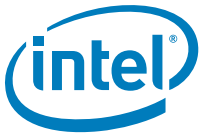
| Master Read/Write Access | | | | |
|---|---|---|---|---|
| **Region** | **Processor and BIOS** | **Intel® ME** | **GbE Controller** | **EC** |
| Descriptor | N/A | N/A | N/A | N/A |
| BIOS | Processor and BIOS can always read from and write to BIOS Region | Read/Write | Read/Write | Read/Write |
|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

### 32.7.1.4 Flash Access

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

#### 32.7.1.4.1 Direct Access

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



**Direct Access Security**

- Requester ID of the device must match that of the primary Requester ID in the 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
  - Supports the same cache flush mechanism in ICH7 which includes Program Register Writes.

#### 32.7.1.4.2 Program Register Access

- Program Register Accesses are not allowed to cross a 4-KB boundary and can not issue a command that might extend across two components
- Software programs the FLA corresponding to the region desired
  - Software must read the devices Primary Region Base/Limit address to create a FLA.

**Register Access Security**

- Only primary region masters can access the registers

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

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

## 32.7.2 SPI Support for TPM

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

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

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

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

§ §

# 33 Testability

## 33.1 JTAG

### 33.1.1 Acronyms

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

### 33.1.2 References

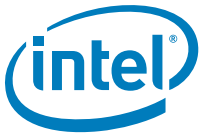
| Specification | Location |
|---|---|
| IEEE Standard Test Access Port and Boundary Scan Architecture | http://standards.ieee.org/findstds/standard/1149.1-2013.html |

### 33.1.3 Overview

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

### 33.1.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **JTAG_TCK** | I/O | **Test Clock Input (TCK):** The test clock input provides the clock for the JTAG test logic. |
| **JTAG_TMS** | I/OD | **Test Mode Select (TMS):** The signal is decoded by the Test Access Port (TAP) controller to control test operations. |
| **JTAG_TDI** | I/OD | **Test Data Input (TDI):** Serial test instructions and data are received by the test logic at TDI. |
| **JTAG_TDO** | I/OD | **Test Data Output (TDO):** TDO is the serial output for test instructions and data from the test logic defined in this standard. |
| **JTAGX** | I/O | This pin is used to support merged debug port topologies. |
| **ITP_PMODE** | O | This signal is used to transmit processor and PCH power/reset information to the ITP Debugger. |
| **PCH_TRIGIN** | I | From CPU, for cross die triggering for debug trace |
| **PCH_TRIGOUT** | O | To CPU IOT for cross die triggering |
| **PREQ#** | I/ OD | From PCH to CPU run control by DCI for closed chassis testing |



| Name | Type | Description |
|---|---|---|
| **PRDY#** | I/ OD | Acknowledge from CPU for run control |
| **CPU_TRST#** | O | JTAG output from DCI to CPU |

### 33.1.5 I/O Signal Planes and States

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| **JTAG_TCK** | Primary | Internal PD | Internal PD | Internal PD | Off |
| **JTAG_TMS** | Primary | Internal PU | Internal PU | Internal PU | Off |
| **JTAG_TDI** | Primary | Internal PU | Internal PU | Internal PU | Off |
| **JTAG_TDO** | Primary | Undriven | Undriven | Undriven | Off |
| **JTAGX**[1] | Primary | Internal PU (as TDO)/Internal PD (as TCK) | Internal PU/ Internal PD | Internal PU/ Internal PD | Off |
| **ITP_PMODE**[2] | Primary | Internal PU | Internal PU | Internal PU | Off |
| **PCH_TRIGIN** | Primary | Internal PD | Internal PD | Undriven | Off |
| **PCH_TRIGOUT** | Primary | Internal PD | Internal PD | Undriven | Off |
| **PREQ#** | Primary | Internal PU | Internal PU | Undriven | Off |
| **PRDY#** | Primary | Internal PU | Internal PU | Undriven | Off |
| **CPU_TRST#** | Primary | Internal PD | Internal PD | Internal PD | Off |

***Notes:***

1. This signal is used in common JTAG topology to take in last device's TDO to DCI. The only planned supported topology is the Shared Topology. Thus, this pin will operate as TCK mode.
2. This pin is connected to HOOK[6] on the merged debug topology.

## 33.2 Intel® Trace Hub (Intel® TH)

### 33.2.1 Overview

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

Intel® Trace Hub is a set of silicon features with supported software API. The primary purpose is to collect trace data from different sources in the system and combine them into a single output stream with time-correlated to each other. Intel® Trace Hub uses common hardware interface for collecting time-correlated system traces through standard destinations. Intel® Trace Hub adopts industry standard (MIPI* STPv2) debug methodology for system debug and software development.

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

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

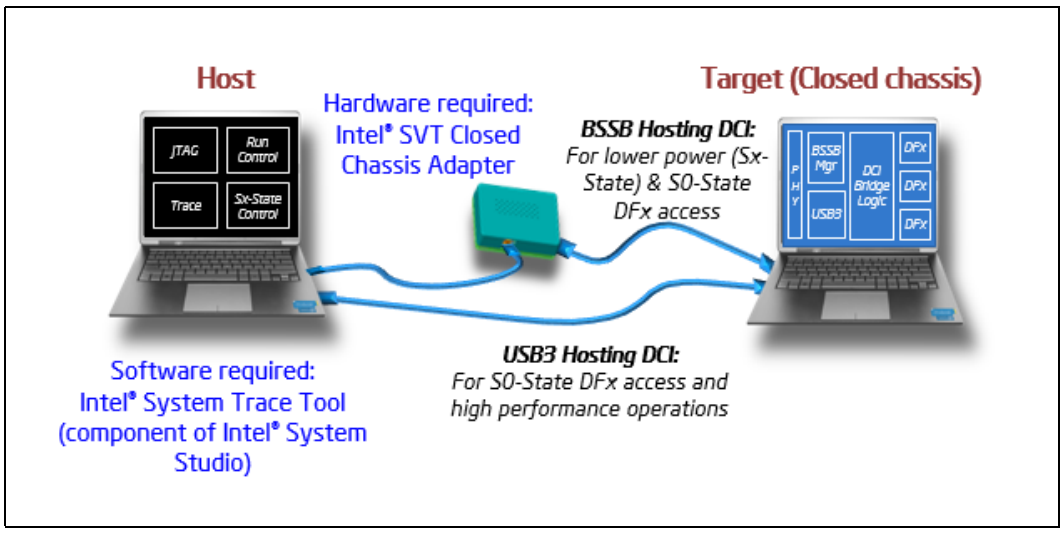
- System Memory

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

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

## 33.2.2 Platform Setup

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




# 33.3 Direct Connect Interface (DCI)

Direct Connect Interface (DCI) is a new debug transport technology to enable closed chassis debug through any of USB3 ports out from Intel silicon. Some bridging logic is embedded in the silicon to "bridge" the gap between standard I/O ports and the debug interfaces including JTAG, probe mode, hooks, trace infrastructure, and etc. To control the operation of this embedded logic, a DCI packet based protocol is invented which controls and data can be sent or received. This protocol can operate over a few different physical transport paths to the target which known as "hosting interfaces".

***Note:*** DCI and USB based debugger (kernel level debugger) are mutually exclusive.

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

- BSSB Hosting DCI
- USB3 Hosting DCI

Supported capabilities in DCI are:

- Closed Chassis Debug at S0 & Sx State
- JTAG Access & Run Control (Probe Mode)

- System Tracing with Intel® Trace Hub

Debug host software that support DCI are:

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

### 33.3.1 Boundary Scan Side Band (BSSB) Hosting DCI

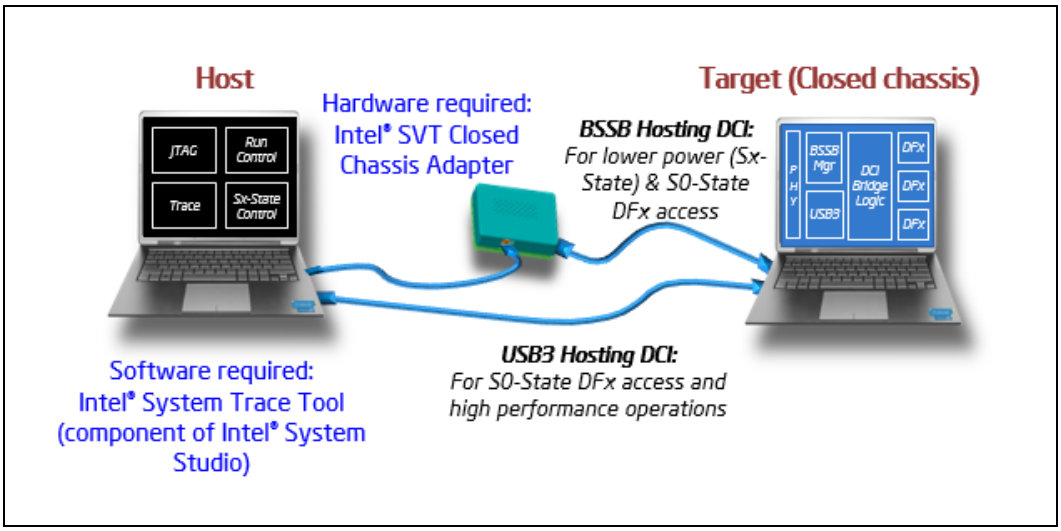
BSSB was developed to provide an alternate path to convey controls and data to or from the embedded logic by connecting physically to the target through a USB3 port. BSSB provides an alternate side band path around the USB3 controller, so that the embedded logic can be accessed, even when the USB controller is not alive (such as in low power states), or is malfunctioning. This path does not rely on USB protocol, link layer, or physical layer, because the xHCI functions are generally not available in such conditions. Instead, this path relies on a special adapter that developed by Intel called Intel® SVT Closed Chassis Adapter (CCA). It is a simple data transformation device. This adapter generates a BSSB signaling protocol operating at up to 400 MHz and serializes data flowing through it. This adapter works together with debug host software and the embedded logic, contain a back-pressure scheme that makes both sides tolerant of overflow and starvation conditions, which is the moral equivalent of the USB link layer. This path also uses native DCI packet protocol instead of USB protocol.

### 33.3.2 USB3 Hosting DCI

It relies on Debug Class Devices (DbC) which is 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 link layer flow control and USB3 physical layer at 5 GHz.

### 33.3.3 Platform Setup

**Figure 33-2. Platform Setup with DCI Connection**




§ §

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

## 34.1 Acronyms

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

## 34.2 References

None

## 34.3 Overview

The PCH implements three independent UART interfaces, UART0, UART1 and UART2. Each UART interface is a 4-wire interface supporting up to 6.25 Mbit/s.

The interfaces can be used in the low-speed, full-speed, and high-speed modes. The UART communicates with serial data ports that conform to the RS-232 interface protocol.

UART2 only implements the UART Host controller and does not incorporate a DMA controller which is implemented for UART0 and UART1. Therefore, UART2 is restricted to operate in PIO mode only

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

## 34.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **UART0_RXD**/ GPP_C8 | I | **UART 0 Receive Data** |
| **UART0_TXD**/ GPP_C9 | O | **UART 0 Transmit Data** |
| **UART0_RTS#**/ GPP_C10 | O | **UART 0 Request to Send** |
| **UART0_CTS#**/ GPP_C11 | I | **UART 0 Clear to Send** |
| **UART1_RXD**/ ISH_UART1_RXD/ GPP_C12 | I | **UART 1 Receive Data** |
| **UART1_TXD**/ ISH_UART1_TXD/ GPP_C13 | O | **UART 1 Transmit Data** |

| Name | Type | Description |
|---|---|---|
| **UART1_RTS#**/ ISH_UART1_RTS#/ GPP_C14 | O | **UART 1 Request to Send** |
| **UART1_CTS#**/ ISH_UART1_CTS#/ GPP_C15 | I | **UART 1 Clear to Send** |
| **UART2_RXD**/ GPP_C20 | I | **UART 2 Receive Data** |
| **UART2_TXD**/ GPP_C21 | O | **UART 2 Transmit Data** |
| **UART2_RTS#**/ GPP_C22 | O | **UART 2 Request to Send** |
| **UART2_CTS#**/ GPP_C23 | I | **UART 2 Clear to Send** |

## 34.5 Integrated Pull-Ups and Pull-Downs

None

## 34.6 I/O Signal Planes and States

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| **UART[2:0]_RXD** | Primary | Undriven | Undriven | Undriven | Off |
| **UART[2:0]_TXD** | Primary | Undriven | Undriven | Undriven | Off |
| **UART[2:0]_RTS#** | Primary | Undriven | Undriven | Undriven | Off |
| **UART[2:0]_CTS#** | Primary | Undriven | Undriven | Undriven | Off |

## 34.7 Functional Description

### 34.7.1 Features

The UART interfaces support the following features:

- Up to 6.25 Mbits/s Auto Flow Control mode as specified in the 16750 standard
- Transmitter Holding Register Empty (THRE) interrupt mode
- 64-byte TX and 64-byte RX host controller FIFOs
- DMA support with 64-byte DMA FIFO per channel (up to 32-byte burst)
- Functionality based on the 16550 industry standards
- Programmable character properties, such as number of data bits per character (5-8), optional parity bit (with odd or even select) and number of stop bits (1, 1.5, or 2)
- Line break generation and detection
- DMA signaling with two programmable modes
- Prioritized interrupt identification
- Programmable FIFO enable/disable

- Programmable serial data baud rate
- Modem and status lines are independently controlled
- Programmable BAUD RATE supported (baud rate = (serial clock frequency)/(16xdivisor))

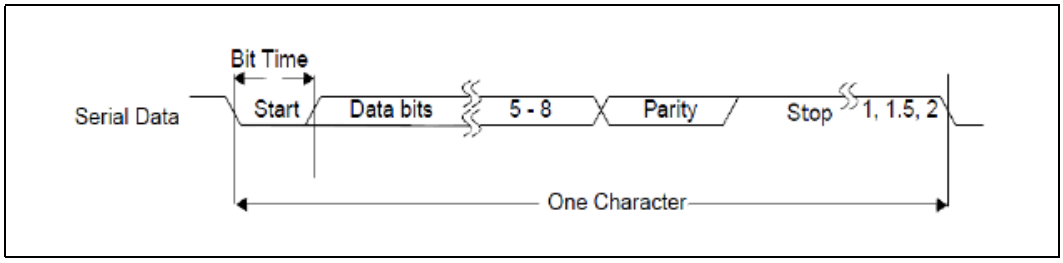
***Notes:***

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

## 34.7.2 UART Serial (RS-232) Protocols Overview

Because the serial communication between the UART host controller and the selected device is asynchronous, Start and Stop bits are used on the serial data to synchronize the two devices. The structure of serial data accompanied by Start and Stop bits is referred to as a character.

**Figure 34-1. UART Serial Protocol**



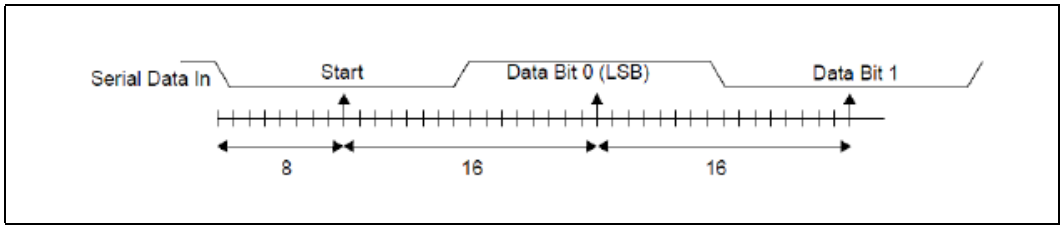

The UART Host Controller Line Control Register (LCR) is used to control the serial character characteristics. The individual bits of the data word are sent after the Start bit, starting with the least significant bit (LSB). These are followed by the optional parity bit, followed by the Stop bit(s), which can be 1, 1.5, or 2.

The Stop bit duration implemented by UART host controller may appear longer due to idle time inserted between characters for some configurations and baud clock divisor values in the transmit direction.

All bit in the transmission (with exception to the half stop bit when 1.5 stop bits are used) are transmitted for exactly the same time duration (which is referred to as Bit Period or Bit Time). One Bit Time equals to 16 baud clocks.

To ensure stability on the line, the receiver samples the serial input data at approximately the midpoint of the Bit Time once the start bit has been detected.

**Figure 34-2. UART Receiver Serial Data Sample Points**



### 34.7.3 16550 8-bit Addressing - Debug Driver Compatibility

The PCH UART host controller is not compatible with legacy UART 16550 debug-port drivers. The UART host controller operates in 32-bit addressing mode only. UART 16550 legacy drivers only operate with 8-bit (byte) addressing. In order to provide compatibility with standard in-box legacy UART drivers a 16550 Legacy Driver mode has been implemented in the UART controller that will convert 8-bit addressed accesses from the 16550 legacy driver to the 32-bit addressing that the UART host controller supports. The control of this mode is via the GEN_REGGRW7 register (UART Additional Registers, offset 0x618h). Refer to register section for the description of these bits.

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

### 34.7.4 DMA Controller

The UART controllers 0 and 1 (UART0 and UART1) have an integrated DMA controller. Each channel contains a 64-byte FIFO. Max. burst size supported is 32 bytes.

UART controller 2 (UART2) only implements the host controllers and does not incorporate a DMA. Therefore, UART2 is restricted to operate in PIO mode only.

#### 34.7.4.1 DMA Transfer and Setup Modes

The DMA can operate in the following modes:

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

The DMA supports the following modes for programming:

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

#### 34.7.4.2 Channel Control

- The source transfer width and destination transfer width are programmed. It can vary to 1 byte, 2 bytes, and 4 bytes.
- Burst size is configurable per channel for source and destination. The number is a power of 2 and can vary between 1,2,4,...,128. This number times the transaction width gives the number of bytes that will be transferred per burst.
- Individual Channel enables. If the channel is not being used, then it should be clock gated.
- Programmable Block size and Packing/Unpacking. Block size of the transfer is programmable in bytes. The block size is not be limited by the source or destination transfer widths.

- Address incrementing modes: The DMA has a configurable mechanism for computing the source and destination addresses for the next transfer within the current block. The DMA supports incrementing addresses and constant addresses.
- Flexibility to configure any hardware handshake sideband interface to any of the DMA channels.
- Early termination of a transfer on a particular channel.

## 34.7.5 Reset

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

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

## 34.7.6 Power Management

### 34.7.6.1 Device Power Down Support

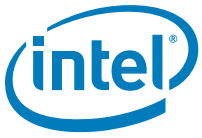
In order to power down peripherals connected to PCH UART bus, the idle, configured state of the I/O signals must be retained to avoid transitions on the bus that can affect the connected powered peripheral. Connected devices are allowed to remain in the D0 active or D2 low power states when the bus is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

### 34.7.6.2 Latency Tolerance Reporting (LTR)

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

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

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



### 34.7.7 Interrupts

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

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

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

### 34.7.8 Error Handling

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

§ §

# 35 Universal Serial Bus (USB)

## 35.1 Acronyms

| Acronyms | Description |
|---|---|
| xHCI | eXtensible Host Controller Interface |

## 35.2 References

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

## 35.3 Overview

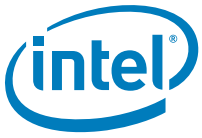
The PCH implements an xHCI USB controller which provides support for up to 14 USB 2.0 signal pairs and 10 SuperSpeed USB 3.0 signal pairs. The xHCI controller supports wake up from sleep states S1-S4. The xHCI USB controller supports up to 64 devices and 128 endpoints.

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

***Note:*** EHCI is no longer supported in PCH.

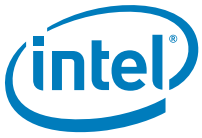
## 35.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **USB3_1_RXN, USB3_1_RXP** | I | **USB 3.0 Differential Receive Pair 1:** These are USB 3.0-based high-speed differential signals for Port #1 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). This port also supports Dual Role Capability for USB On The Go. |
| **USB3_1_TXN, USB3_1_TXP** | O | **USB 3.0 Differential Transmit Pair 1:** These are USB 3.0-based high-speed differential signals for Port #1 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). This port also supports Dual Role Capability for USB On The Go. |
| **USB3_2_RXN, USB3_2_RXP**/ | I | **USB 3.0 Differential Receive Pair 2:** These are USB 3.0-based high-speed differential signals for Port #2 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). |
| **USB3_2_TXN, USB3_2_TXP**/ | O | **USB 3.0 Differential Transmit Pair 2:** These are USB 3.0-based high-speed differential signals for Port #2 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). |
| **USB3_3_RXN**, **USB3_3_RXP** | I | **USB 3.0 Differential Receive Pair 3:** These are USB 3.0-based high-speed differential signals for Port #3 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). |
| **USB3_3_TXN, USB3_3_TXP** | O | **USB 3.0 Differential Transmit Pair 3:** These are USB 3.0-based high-speed differential signals for Port #3 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). |



| Name | Type | Description |
|---|---|---|
| **USB3_4_RXN, USB3_4_RXP** | I | **USB 3.0 Differential Receive Pair 4:** These are USB 3.0-based high-speed differential signals for Port #4 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). |
| **USB3_4_TXN, USB3_4_TXP** | O | **USB 3.0 Differential Transmit Pair 4:** These are USB 3.0-based high-speed differential signals for Port #4 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). |
| **USB3_5_RXN, / USB3_5_RXP/** | I | **USB 3.0 Differential Receive Pair 5:** These are USB 3.0-based high-speed differential signals for Port #5 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). |
| **USB3_5_TXN, / USB3_5_TXP/** | O | **USB 3.0 Differential Transmit Pair 5:** These are USB 3.0-based high-speed differential signals for Port #5 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). |
| **USB3_6_RXN, / USB3_6_RXP/** | I | **USB 3.0 Differential Receive Pair 6:** These are USB 3.0-based high-speed differential signals for Port #6 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). |
| **USB3_6_TXN, / USB3_6_TXP/** | O | **USB 3.0 Differential Transmit Pair 6:** These are USB 3.0-based high-speed differential signals for Port #6 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent). |
| **USB3_7_RXN /** PCIE1_RXN, **USB3_7_RXP /** PCIE1_RXP | I | **USB 3.0 Differential Receive Pair 7:** These are USB 3.0-based high-speed differential signals for Port #7 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent).<br>***Note:*** Use FITC to set the soft straps that select this port as PCIe Port 1. |
| **USB3_7_TXN /** PCIE1_TXN, **USB3_7_TXP /** PCIE1_TXP | O | **USB 3.0 Differential Transmit Pair 7:** These are USB 3.0-based high-speed differential signals for Port #7 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent).<br>***Note:*** Use FITC to set the soft straps that select this port as PCIe Port 1. |
| **USB3_8_RXN /** PCIE2_RXN, **USB3_8_RXP /** PCIE2_RXP | I | **USB 3.0 Differential Receive Pair 8:** These are USB 3.0-based high-speed differential signals for Port #8 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent).<br>***Note:*** Use FITC to set the soft straps that select this port as PCIe Port 2. |
| **USB3_8_TXN /** PCIE2_TXN, **USB3_8_TXP /** PCIE2_TXP | O | **USB 3.0 Differential Transmit Pair 8:** These are USB 3.0-based high-speed differential signals for Port #8 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent).<br>***Note:*** Use FITC to set the soft straps that select this port as PCIe Port 2. |
| **USB3_9_RXN /** PCIE3_RXN, **USB3_9_RXP /** PCIE3_RXP | I | **USB 3.0 Differential Receive Pair 9:** These are USB 3.0-based high-speed differential signals for Port #9 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent).<br>***Note:*** Use FITC to set the soft straps that select this port as PCIe Port 3. |
| **USB3_9_TXN /** PCIE3_TXN, **USB3_9_TXP /** PCIE3_TXP | O | **USB 3.0 Differential Transmit Pair 9:** These are USB 3.0-based high-speed differential signals for Port #9 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent).<br>***Note:*** Use FITC to set the soft straps that select this port as PCIe Port 3. |
| **USB3_10_RXN /** PCIE4_RXN, **USB3_10_RXP /** PCIE4_RXP | I | **USB 3.0 Differential Receive Pair 10:** These are USB 3.0-based high-speed differential signals for Port #10 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent).<br>***Note:*** Use FITC to set the soft straps that select this port as PCIe Port 4. |
| **USB3_10_TXN /** PCIE4_TXN, **USB3_10_TXP /** PCIE4_TXP | O | **USB 3.0 Differential Transmit Pair 10:** These are USB 3.0-based high-speed differential signals for Port #10 and the xHCI Controller. It should map to a USB connector with one of the OC (overcurrent).<br>***Note:*** Use FITC to set the soft straps that select this port as PCIe Port 4. |
| **USB2P_1, USB2N_1** | I/O | **USB 2.0 Port 1 Transmit/Receive Differential Pair 1:**<br>This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. This port also supports Dual Role Capability for USB On The Go. |

| Name | Type | Description |
|---|---|---|
| **USB2P_2, USB2N_2** | I/O | **USB 2.0 Port 2 Transmit/Receive Differential Pair 2:** This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. |
| **USB2P_3, USB2N_3** | I/O | **USB 2.0 Port 3Transmit/Receive Differential Pair 3:** This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. |
| **USB2P_4, USB2N_4** | I/O | **USB 2.0 Port 4 Transmit/Receive Differential Pair 4:** This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. |
| **USB2P_5, USB2N_5** | I/O | **USB 2.0 Port 5 Transmit/Receive Differential Pair 5:** This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. |
| **USB2P_6, USB2N_6** | I/O | **USB 2.0 Port 6 Transmit/Receive Differential Pair 6:** This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. |
| **USB2P_7, USB2N_7** | I/O | **USB 2.0 Port 7 Transmit/Receive Differential Pair 7:** This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. |
| **USB2P_8, USB2N_8** | I/O | **USB 2.0 Port 8 Transmit/Receive Differential Pair 8:** This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. |
| **USB2P_9, USB2N_9** | I/O | **USB 2.0 Port 9 Transmit/Receive Differential Pair 9:** This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. |
| **USB2P_10, USB2N_10** | I/O | **USB 2.0 Port 10 Transmit/Receive Differential Pair 10:** This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. |
| **USB2P_11, USB2N_11** | I/O | **USB 2.0 Port 11 Transmit/Receive Differential Pair 11:** This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. |
| **USB2P_12, USB2N_12** | I/O | **USB 2.0 Port 12 Transmit/Receive Differential Pair 12:** This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. |
| **USB2p_13, USB2n_13** | I/O | **USB 2.0 Port 13 Transmit/Receive Differential Pair 13:** This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. |
| **USB2P_14, USB2N_14** | I/O | **USB 2.0 Port 14 Transmit/Receive Differential Pair 14:** This USB 2.0 signal pair are routed to xHCI Controller and should map to a USB connector with one of the overcurrent OC. |
| **USB2_OC0**#/ GPP_E9 | I | **Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred. |
| **USB2_OC1**# / GPP_E10 | I | **Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred. |
| **USB2_OC2#/** GPP_E11 | I | **Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred. |
| **USB2_OC3#/** GPP_E12 | I | **Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred. |
| **USB2_OC4**# / GPP_F15 | I | **Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred. |
| **USB2_OC5#** / GPP_F16 | I | **Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred. |
| **USB2_OC6# /** GPP_F17 | I | **Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred. |
| **USB2_OC7# /** GPP_F18 | I | **Overcurrent Indicators**: These signals set corresponding bits in the USB controller to indicate that an overcurrent condition has occurred. |



| Name | Type | Description |
|---|---|---|
| **USB2_VBUSSENSE** | I | VBUS Sense for USB On-The-Go. Refer to OTG 2.0 specification for the sensing threshold voltage spec. |
| **USB2_ID** | I | ID detect for USB On The Go. |
| **USB2_COMP** | I | USB Resistor Bias, analog connection points for an external resistor to ground. |

## 35.5 Integrated Pull-Ups and Pull-Downs

| Signal | Resistor Type | Value | Notes |
|---|---|---|---|
| **USB2N_[14:1]** | Internal Pull-down | 14.25–24.8K | 1 |
| **USB2P_[14:1]** | Internal Pull-down | 14.25–24.8K | 1 |
| **USB2_ID** | Internal Weak Pull-up | | If this signal is not in use, then it should have 1k PD to ground. |
| ***Note:***<br>1. Series resistors (45 ohm ±10%) | | | |

## 35.6 I/O Signal Planes and States

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| **USB3_[10:1]_RXN**<br>**USB3_[10:1]_RXP** | Primary | Internal Pull-down | Internal Pull-down | Internal Pull-down | OFF |
| **USB3_[10:1]_TXN**<br>**USB3_[10:1]_TXP** | Primary | Internal Pull-down | Internal Pull-down | Internal Pull-down | OFF |
| **USB2N_[14:1]** | DSW | Internal Pull-down | Internal Pull-down | Internal Pull-down | Internal Pull-down |
| **USB2P_[14:1]** | DSW | Internal Pull-down | Internal Pull-down | Internal Pull-down | Internal Pull-down |
| **USB2_OC0#** | Primary | Undriven | Undriven | Undriven | OFF |
| **USB2_OC1#** | Primary | Undriven | Undriven | Undriven | OFF |
| **USB2_OC2#** | Primary | Undriven | Undriven | Undriven | OFF |
| **USB2_OC3#** | Primary | Undriven | Undriven | Undriven | OFF |
| **USB2_OC4#** | Primary | Undriven | Undriven | Undriven | OFF |
| **USB2_OC5#** | Primary | Undriven | Undriven | Undriven | OFF |
| **USB2_OC6#** | Primary | Undriven | Undriven | Undriven | OFF |
| **USB2_OC7#** | Primary | Undriven | Undriven | Undriven | OFF |
| **USB2_VBUSSENSE** | Primary | Undriven | Undriven | Undriven | OFF |
| **USB2_ID**[1] | Primary | Internal Pull-UP | Undriven/ Internal Pull-UP | Undriven/ Internal Pull-UP | OFF |
| **USB2_COMP** | Primary | Undriven | Undriven | Undriven | OFF |

***Notes:***

1. The USB2_ID pin is pulled up internally.

# 35.7 Functional Description

## 35.7.1 eXtensible Host Controller Interface (xHCI) Controller (D20:F0)

The PCH contains an eXtensible Host Controller Interface (xHCI) host controller which supports up to 14–USB 2.0 ports and up to 10 –USB 3.0 ports with board routing, ACPI table and BIOS considerations. This controller allows data transfers of up to 5Gb/s. The controller supports SuperSpeed (SS), High-Speed (HS), Full-Speed (FS) and Low-Speed (LS) traffic on the bus. The xHCI controller supports USB Debug port on all USB 3.0-capable ports. The xHCI also supports USB Attached SCIS Protocol (UASP).

The PCH also supports Dual Role Capability. The USB Host Controller can now be paired with a standalone USB device to provide dual role functionality. The USB subsystem incorporates a USB 3.0 device controller. This controller is instantiated as a separate PCI function and shares USB 2.0 port 1 and USB 3.0 port 1. The PCH USB implementation is compliant to the Device specification and supports host/device control through ID pin. The ID pin is an input micro AB connector and signifies the type of agent connected to the port.

### 35.7.1.1 USB Dual Role Support

The Device controller shares USB 3.0 port #1 and USB 2.0 port #1 with the host controller, with ownership of the port being decided by the ID pin. A 1 on the ID pin signifies that the port is to be mapped to the device controller. A 0 signifies that the port is to be mapped to the host controller. While the port is mapped to the device controller the host controller Rx detection must always indicate a disconnected port.

§ §

# 36 GPIO Serial Expander

## 36.1 Acronyms, Definitions

| Acronyms | Description |
|---|---|
| GSX | GPIO Serial Expander |

## 36.2 References

None

## 36.3 Overview

GPIO Serial Expander (GSX) is the capability provided by the PCH to expand the GPIOs on a platform that needs more GPIOs than the ones provided by the PCH. The solution requires external shift register discrete components.

## 36.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **GSXDOUT** / GPP_G12 | O | GPIO Serial Expander Controller Data Out |
| **GSXSLOAD** / GPP_G13 | O | GPIO Serial Expander Controller Serial Load |
| **GSXDIN /** GPP_G14 | I | GPIO Serial Expander Controller Data In |
| **GSXSRESET#** / GPP_G15 | O | GPIO Serial Expander Controller Serial Reset |
| **GSXCLK** / GPP_G16 | O | GPIO Serial Expander Controller Clock |

## 36.5 Integrated Pull-ups and Pull-downs

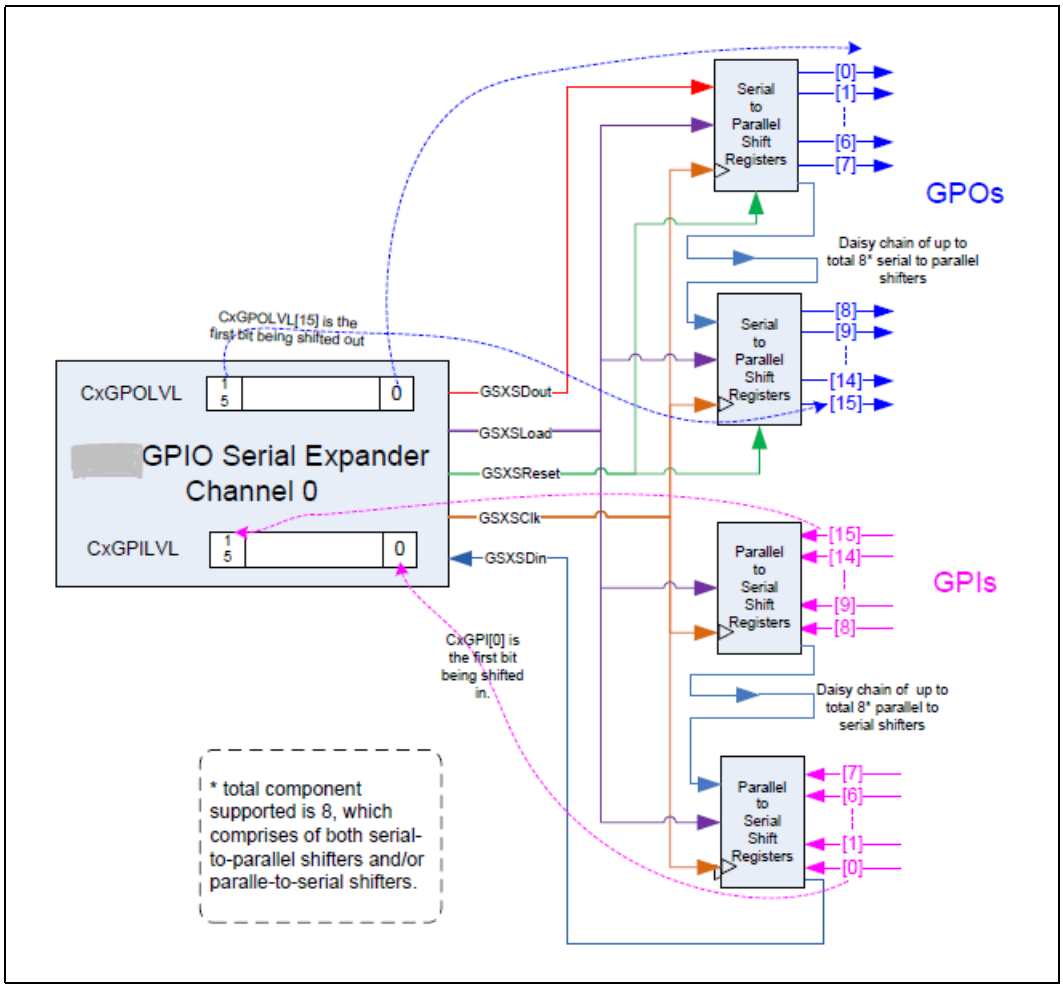
None

## 36.6 Functional Description

GPIO Serial Expander (GSX) uses serial-to-parallel or parallel-to-serial shift register discrete components to increase number of the GPIO pins for system use. It expands in the multiples of 8 for input or output with 8 pins per expander. The total shift register component supported is 8, which can expand the GPIOs by up to 64.

Figure 36-1 illustrates a GPIO expansion topology with 16 GPIs and 16 GPOs.

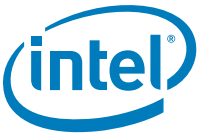
**Figure 36-1. Example of GSX Topology**




Coming out of system reset, GSX is in reset with the following behaviors:

- GSXSRESET# asserted by default. The signal remains asserted until BIOS/SW initialization has been completed and CxCMD.ST set to 1.
- GSXSLOAD is 0 by default until CxCMD.ST is set to 1.
- GSXSCLK is not toggling until CxCMD.ST is set to 1.

§ §



# 37 Fan Control

## 37.1 Acronyms

| Acronyms | Description |
|---|---|
| PWM | Pulse Width Modulation |
| FSC | Fan Speed Controller |

## 37.2 References

None.

## 37.3 Overview

The Fan Speed Controller is used to control the fans in the system.

## 37.4 Signal Description

| Name | Type | Description |
|---|---|---|
| **FAN_TACH_0** / GPP_G0 | I | Tachometer pulse input signal |
| **FAN_TACH_1** / GPP_G1 | I | Tachometer pulse input signal |
| **FAN_TACH_2** / GPP_G2 | I | Tachometer pulse input signal |
| **FAN_TACH_3** / GPP_G3 | I | Tachometer pulse input signal |
| **FAN_TACH_4** / GPP_G4 | I | Tachometer pulse input signal |
| **FAN_TACH_5** / GPP_G5 | I | Tachometer pulse input signal |
| **FAN_TACH_6** / GPP_G6 | I | Tachometer pulse input signal |
| **FAN_TACH_7** / GPP_G7 | I | Tachometer pulse input signal |
| **FAN_PWM_0** / GPP_G8 | OD | Pulse Width Modulated duty cycle output |
| **FAN_PWM_1** / GPP_G9 | OD | Pulse Width Modulated duty cycle output |
| **FAN_PWM_2** / GPP_G10 | OD | Pulse Width Modulated duty cycle output |
| **FAN_PWM_3** / GPP_G11 | OD | Pulse Width Modulated duty cycle output |

## 37.5 Integrated Pull-ups and Pull-downs

| Signal | Resistor Type | Value | Notes |
|---|---|---|---|
| FAN_TACH_[7:0] | Internal Pull-up | 14K - 26K | |

## 37.6 I/O Signal Planes and States

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| FAN_TACH_[7:0] | Primary | Undriven | Undriven | Undriven | Off |
| FAN_PWM_[3:0] | Primary | Undriven | Undriven | Undriven | Off |

## 37.7 Functional Description

The Fan Speed Controller (FSC) is used to control fans in the system. FSC drives four pulse width modulated signals to control the speed of the fans. FSC samples eight tachometer inputs to measure the speed of the fans. FSC also has an interface to the on-die thermal sensor to know the current temperature.

PWM signal is driven as open-drain. An external pull-up resistor should be implemented to provide the rising edge of the PWM output signal. The PWM output is driven low during reset, which represents 0% duty cycle to the majority of fans.

Note that if a PWM output is programmed to inverted polarity for a particular fan, then the low voltage driven during reset represents 100% duty cycle to the fan.

§ §

# 38 Direct Media Interface

## 38.1 Acronyms

| Acronyms | Description |
|---|---|
| DMI | Direct Media Interface |

## 38.2 References

| Specification | Location |
|---|---|
| PCI Express* Specification | http://www.pcisig.com/specifications |

## 38.3 Overview

The PCH communicates with the processor using high-speed DMI that supports 8 GT/s data rates.

## 38.4 Signal Description

| Name | Type | Description |
|---|---|---|
| DMI_RXN[3:0]<br>DMI_RXP[3:0] | I | DMI receive lanes |
| DMI_TXN[3:0]<br>DMI_TXP[3:0] | O | DMI transmit lanes |

## 38.5 Integrated Pull-ups and Pull-downs

| Signal | Resistor Type | Value | Notes |
|---|---|---|---|
| DMI_RXN[3:0]<br>DMI_RXP[3:0] | Pull-down | 14K - 26K | |
| DMI_TXN[3:0]<br>DMI_TXP[3:0] | Pull-down | 14K - 26K | |

***Note:*** Depending on the platform usage, the default is terminated to VSS. If it is terminated to VCC, the default value will be high. DMI_RX*/DMI_TX* pins terminated value are determined by soft-straps. For AC coupling mode, DMI_TX* pins are terminated to VCC/2 and DMI_RX* pins are terminated to VSS.

## 38.6 I/O Signal Planes and States

| Signal Name | Power Plane | During Reset | Immediately after Reset | S3/S4/S5 | Deep Sx |
|---|---|---|---|---|---|
| DMI_RXN[3:0]<br>DMI_RXP[3:0] | Primary | Internal Pull-down | Internal Pull-down | Internal Pull-down | Off |
| DMI_TXN[3:0]<br>DMI_TXP[3:0] | Primary | Internal Pull-down | Internal Pull-down | Internal Pull-down | Off |

## 38.7 Functional Description

PCH DMI is compliant to the DMI 2.0 specification with a bit rate of 2.5GT/s, 5.0GT/s and 8.0GT/s. The DMI Link is compliant to the PCI Express* 3.0 specification for the root complex internal links that support up to 8.0GT/s, with the exceptions called out in the DMI 2.0 specification and in this document.

The DMI supports x4, x2 and x1 link widths through soft straps. The standard PCI Express mechanism for link width negotiation to either x2 or x1 link width change is supported. Some of key features besides PCI Express* Specifications are listed below:

- Addition of LT Memory Write and LT Memory Read TLPs.
- All virtual channels other than VC0 are private and not exposed to the OS.
- Non-unique Transaction IDs are allowed on DMI.
- Downstream requests restricted to VC0 (CPU and peer).
- Shorter than conventional DMI Link Reset sequence.
- DMI is DC coupled by default, but can be set to AC coupled by a strap. Supports half swing voltage on the transmitter.
- DMI can be forced to Detect as either x1, x2 or x4 using soft straps.
- Upstream IO and Configuration cycles are not supported.
- DMI does not implement the PCI Express defined Root Complex Register Block and is not OS visible.

***Note:*** Polarity inversion and lane reversal on DMI link is not allowed.

§ §

# 39 Primary to Sideband Bridge (P2SB)

## 39.1 Overview

The PCH incorporates a wide variety of devices and functions. The registers within these devices are mainly accessed through the primary interface, such as PCI configuration space and IO/MMIO space. Some devices also have registers that are distributed within the PCH Private Configuration Space at individual endpoints (Target Port IDs) which are only accessible through the PCH Sideband Interface.

These PCH Private Configuration Space Registers can be addressed via SBREG_BAR or through SBI Index Data pair programming.

**Table 39-1. Private Configuration Space Register Target Port IDs (Sheet 1 of 2)**

| PCH Device/Function Type | Target Port ID |
|---|---|
| HSIO Strap Configuration | 0x89 |
| General Purpose I/O (GPIO) Community 3 | 0xAC |
| General Purpose I/O (GPIO) Community 2 | 0xAD |
| General Purpose I/O (GPIO) Community 1 | 0xAE |
| General Purpose I/O (GPIO) Community 0 | 0xAF |
| DCI | 0xB8 |
| PSF1- Function Disable | 0xBA |
| PSF2- Function Disable | 0xBB |
| PSF3- Function Disable | 0xBC |
| PSF4- Function Disable | 0xBD |
| ISH Controller | 0xBF |
| Real Time Clock (RTC) | 0xC3 |
| Processor Interface, 8254 Timer, HPET, APIC | 0xC4 |
| SMBus | 0xC6 |
| LPC | 0xC7 |
| USB2.0 | 0xCA |
| UART, I2C, GSPI Interface* | 0xCB |
| FIA Configuration | 0xCF |
| HDA / DSP | 0xD7 |
| SATA | 0xD9 |
| Integrated Clock Controller (ICC) | 0xDC |
| PCIe Controller #1 (SPA) | 0xE0 |
| PCIe Controller #2 (SPB) | 0xE1 |

### Table 39-1. Private Configuration Space Register Target Port IDs (Sheet 2 of 2)

| PCH Device/Function Type | Target Port ID |
|---|---|
| PCIe Controller #3 (SPC) | 0xE2 |
| PCIe Controller #4 (SPD) | 0xE3 |
| PCIe Controller #5 (SPE) | 0xE4 |
| USB Dual Role / OTG | 0xE5 |
| xHCI | 0xE6 |
| MODPHY0 (HSIO Lanes #1 - #6) | 0xEA |
| MODPHY1 (HSIO Lanes #7 - #14) | 0xE9 |
| MODPHY2 (HSIO Lanes #15 - #18) | 0xA9 |
| MODPHY3 (HSIO Lanes 19-26) | 0xA8 |
| eSPI / SPI | 0xEE |
| DMI Configuration | 0xEF |

**Note:** FID[7:0] consists of Device[7:3], Function[2:0] for $I^2C$, UART, GSPI PCI Cfg and MMIO space.

§ §

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Intel:

GLC232 S R2CB GL82CM238 S R30U GLH110 S R2CA GLB150 S R2C7 GLQ150 S R2C6 GL82HM175 S R30W GL82QM175 S R30V GLC236 S R2CC GLZ170 S R2C9

Стандарт Электрон Связь

Мы молодая и активно развивающаяся компания в области поставок электронных компонентов. Мы поставляем электронные компоненты отечественного и импортного производства напрямую от производителей и с крупнейших складов мира.

Благодаря сотрудничеству с мировыми поставщиками мы осуществляем комплексные и плановые поставки широчайшего спектра электронных компонентов.

Собственная эффективная логистика и склад в обеспечивает надежную поставку продукции в точно указанные сроки по всей России.

Мы осуществляем техническую поддержку нашим клиентам и предпродажную проверку качества продукции. На все поставляемые продукты мы предоставляем гарантию .

Осуществляем поставки продукции под контролем ВП МО РФ на предприятия военно-промышленного комплекса России , а также работаем в рамках 275 ФЗ с открытием отдельных счетов в уполномоченном банке. Система менеджмента качества компании соответствует требованиям ГОСТ ISO 9001.

Минимальные сроки поставки, гибкие цены, неограниченный ассортимент и индивидуальный подход к клиентам являются основой для выстраивания долгосрочного и эффективного сотрудничества с предприятиями радиоэлектронной промышленности, предприятиями ВПК и научно-исследовательскими институтами России.

С нами вы становитесь еще успешнее!

**Наши контакты:**

**Телефон:** +7 812 627 14 35

**Электронная почта:** sales@st-electron.ru

**Адрес:** 198099, Санкт-Петербург,
Промышленная ул, дом № 19, литера Н,
помещение 100-Н Офис 331