



# Intel<sup>®</sup> Processor and Intel<sup>®</sup> Core<sup>™</sup> i3 N-Series

Datasheet, Volume 1 of 2

---

*Rev. 001*

*January 2023*



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

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

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

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

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

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

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

Copyright © 2023, Intel Corporation. All rights reserved.

## Contents

---

<b>Revision History</b> .....	<b>16</b>
<b>1.0 Introduction</b> .....	<b>17</b>
1.1 Processor Volatility Statement.....	19
1.2 Package Support.....	19
1.3 Supported Technologies.....	19
1.3.1 API Support (Windows*).....	21
1.4 Power Management Support.....	21
1.4.1 Processor Core Power Management.....	21
1.4.2 System Power Management.....	21
1.4.3 Memory Controller Power Management.....	21
1.4.4 Processor Graphics Power Management.....	22
1.5 Thermal Management Support.....	22
1.6 Ball-out Information.....	22
1.7 Processor Testability.....	23
1.8 Operating Systems Support.....	23
1.9 Terminology and Special Marks.....	23
1.10 Flexible High Speed I/O.....	26
1.10.1 Flexible I/O Lane Selection.....	27
1.11 Related Documents.....	28
<b>2.0 Processor and PCH Device IDs</b> .....	<b>29</b>
2.1 CPUID.....	29
2.2 PCI Configuration Header.....	29
2.3 Processor Device IDs.....	30
2.4 PCH Device and Revision IDs .....	31
<b>3.0 Package Mechanical Specifications</b> .....	<b>34</b>
3.1 Package Mechanical Attributes.....	34
3.2 Package Loading and Die Pressure Specifications.....	34
3.2.1 Die Pressure Specifications.....	34
3.3 Package Storage Specifications.....	35
<b>4.0 Memory Mapping</b> .....	<b>36</b>
4.1 Functional Description.....	36
4.1.1 PCI Devices and Functions.....	36
4.1.2 Fixed I/O Address Ranges.....	36
4.1.3 Variable I/O Decode Ranges.....	39
4.2 Memory Map.....	40
4.2.1 Boot Block Update Scheme.....	42
<b>5.0 Pin Straps</b> .....	<b>44</b>
<b>6.0 Electrical and Thermal Characteristics</b> .....	<b>48</b>
<b>7.0 Technologies</b> .....	<b>49</b>
7.1 Platform Environmental Control Interface (PECI).....	49
7.1.1 PECI Bus Architecture.....	49
7.2 Intel® Virtualization Technology (Intel® VT).....	51

- 7.2.1 Intel® Virtualization Technology (Intel® VT) for Intel® 64 and Intel® Architecture (Intel® VT-x) ..... 52
- 7.2.2 Intel® VT for Directed I/O ..... 54
- 7.2.3 Intel® APIC Virtualization Technology ..... 56
- 7.3 Security Technologies..... 57
  - 7.3.1 Intel® Advanced Encryption Standard New Instructions..... 57
  - 7.3.2 Perform Carry-Less Multiplication Quad Word Instruction (PCLMULQDQ) ..... 58
  - 7.3.3 Intel® Secure Key..... 58
  - 7.3.4 Execute Disable Bit ..... 58
  - 7.3.5 Boot Guard Technology ..... 58
  - 7.3.6 Intel® Supervisor Mode Execution Protection (SMEP)..... 59
  - 7.3.7 Intel® Supervisor Mode Access Protection (SMAP)..... 59
  - 7.3.8 Intel® Secure Hash Algorithm Extensions (Intel® SHA Extensions)..... 59
  - 7.3.9 User Mode Instruction Prevention (UMIP) ..... 60
  - 7.3.10 Read Processor ID (RDPID) ..... 60
  - 7.3.11 Control-flow Enforcement Technology (Intel® CET)..... 60
  - 7.3.12 KeyLocker Technology..... 61
- 7.4 Power and Performance Technologies..... 61
  - 7.4.1 Intel® Smart Cache Technology..... 61
  - 7.4.2 IA Core Level 1 and Level 2 Caches ..... 62
  - 7.4.3 Ring Interconnect..... 63
  - 7.4.4 Power Aware Interrupt Routing (PAIR)..... 63
  - 7.4.5 Enhanced Intel SpeedStep® Technology..... 63
  - 7.4.6 Intel® Turbo Boost Technology 2.0..... 63
  - 7.4.7 Intel® Thermal Velocity Boost..... 64
  - 7.4.8 Intel® Speed Shift Technology ..... 65
  - 7.4.9 Intel® Advanced Vector Extensions 2..... 65
  - 7.4.10 Intel® 64 Architecture x2APIC..... 65
  - 7.4.11 Intel® Dynamic Tuning Technology (DTT) ..... 67
  - 7.4.12 Intel® GNA 3.0..... 67
  - 7.4.13 Cache Line Write Back (CLWB)..... 67
  - 7.4.14 Remote Action Request (RAR)..... 67
  - 7.4.15 User Mode Wait Instructions ..... 68
- 7.5 Debug Technologies ..... 68
  - 7.5.1 Intel® Processor Trace ..... 68
  - 7.5.2 Platform CrashLog..... 68
  - 7.5.3 Telemetry Aggregator..... 69

**8.0 Audio Voice and Speech..... 71**

- 8.1 Feature Overview..... 71
  - 8.1.1 Intel® High Definition Audio (Intel® HD Audio) Controller Capabilities..... 72
  - 8.1.2 Audio DSP Capabilities..... 72
  - 8.1.3 Intel® High Definition Audio Interface Capabilities..... 73
  - 8.1.4 Direct Attached Digital Microphone (PDM) Interface..... 73
  - 8.1.5 USB Audio Offload Support..... 74
  - 8.1.6 I<sup>2</sup>S/PCM Interface..... 74
  - 8.1.7 Intel® Display Audio Interface..... 75
  - 8.1.8 MIPI® SoundWire\* Interface..... 75
- 8.2 Signal Description..... 76
- 8.3 Integrated Pull-Ups and Pull-Downs..... 78
- 8.4 I/O Signal Planes and States..... 79

<b>9.0 Image Processing Unit.....</b>	<b>80</b>
9.1 Platform Imaging Infrastructure.....	80
9.2 Intel® Image Processing Unit (Intel® IPU6).....	80
9.3 Camera/MIPI.....	81
9.3.1 Camera Pipe Support.....	81
9.3.2 MIPI* CSI-2 Camera Interconnect.....	81
<b>10.0 Power Management.....</b>	<b>83</b>
10.1 Signal Description.....	85
10.2 Integrated Pull-Ups and Pull-Downs.....	88
10.3 I/O Signal Planes and States.....	88
10.4 Functional Description.....	90
10.4.1 Features.....	90
10.4.2 PCH S0 Low Power.....	90
10.4.3 Power Management Sub-state.....	91
10.4.4 PCH and System Power States.....	91
10.4.5 SMI#/SCI Generation.....	94
10.4.6 C-States.....	96
10.4.7 Sleep States.....	96
10.4.8 Event Input Signals and Their Usage.....	102
10.4.9 ALT Access Mode.....	106
10.4.10 System Power Supplies, Planes, and Signals.....	108
10.4.11 Legacy Power Management Theory of Operation.....	110
10.4.12 Reset Behavior.....	111
10.5 Advanced Configuration and Power Interface (ACPI) States Supported.....	113
10.6 Processor IA Core Power Management.....	114
10.6.1 OS/HW Controlled P-states.....	115
10.6.2 Low-Power Idle States.....	115
10.6.3 Requesting the Low-Power Idle States.....	116
10.6.4 Processor IA Core C-State Rules.....	116
10.6.5 Package C-States.....	117
10.6.6 Package C-States and Display Resolutions.....	120
10.7 Processor Graphics Power Management .....	121
10.7.1 Memory Power Savings Technologies.....	121
10.7.2 Display Power Savings Technologies.....	121
10.7.3 Processor Graphics Core Power Savings Technologies.....	122
10.8 System Agent Enhanced Intel SpeedStep® Technology.....	123
10.9 Type C Sub System (TCSS) Power State.....	123
<b>11.0 Power Delivery.....</b>	<b>125</b>
11.1 Power and Ground Signals.....	125
11.2 FIVR.....	126
11.3 PCH Platform Voltage Rails.....	126
<b>12.0 Thermal Management.....</b>	<b>129</b>
12.1 Processor Thermal Management.....	129
12.1.1 Thermal Considerations.....	129
12.1.2 Assured Power (cTDP) and Low Power Mode (LPM).....	132
12.1.3 Thermal Management Features.....	134
12.1.4 Intel® Memory Thermal Management .....	141
12.2 Processor Line Thermal and Power Specifications.....	141

12.2.1 Processor Line Thermal and Power.....	142
12.3 Error and Thermal Protection Signals.....	143
<b>13.0 PCH Thermal Sensor.....</b>	<b>145</b>
13.1 Modes of Operation.....	145
13.2 Temperature Trip Point.....	145
13.3 Thermal Sensor Accuracy (T <sub>accuracy</sub> ).....	145
13.4 Thermal Reporting to an EC.....	146
13.5 Thermal Trip Signal (PCHHOT#).....	146
<b>14.0 System Clocks.....</b>	<b>147</b>
14.1 ICC.....	147
14.1.1 Signal Descriptions.....	147
14.2 I/O Signal Pin States.....	148
14.3 Clock Topology.....	148
14.3.1 Integrated Reference Clock PLL.....	149
<b>15.0 Real Time Clock (RTC).....</b>	<b>150</b>
15.1 Signal Description.....	150
15.2 I/O Signal Planes and States.....	151
<b>16.0 Memory.....</b>	<b>152</b>
16.1 Signal Description.....	152
16.2 System Memory Interface.....	154
16.2.1 DDR Support Matrix.....	154
16.2.2 Supported Memory Modules and Devices.....	155
16.2.3 System Memory Timing Support.....	157
16.2.4 SAGV Points.....	158
16.2.5 Memory Controller (MC).....	158
16.2.6 Technology Enhancements of Intel® Fast Memory Access (Intel® FMA).....	158
16.2.7 Data Scrambling.....	159
16.2.8 Data Swapping .....	159
16.2.9 Ascending and Descending.....	159
16.2.10 DRAM Clock Generation .....	160
16.2.11 DRAM Reference Voltage Generation .....	160
16.2.12 Data Swizzling.....	160
16.2.13 Error Correction With Standard RAM.....	160
16.3 Integrated Memory Controller (IMC) Power Management.....	160
16.3.1 Disabling Unused System Memory Outputs.....	160
16.3.2 DRAM Power Management and Initialization.....	161
16.3.3 DDR Electrical Power Gating.....	163
16.3.4 Power Training.....	163
<b>17.0 USB-C* Sub System (TCSS).....</b>	<b>164</b>
17.1 General Capabilities.....	164
17.2 USB-C Sub-system xHCI/xDCI Controllers .....	165
17.2.1 USB 3 Controllers.....	165
17.3 USB-C Sub-System Display Interface.....	166
17.4 USB Type-C Signals.....	166
<b>18.0 Universal Serial Bus (USB).....</b>	<b>167</b>
18.1 Functional Description.....	167
18.1.1 eXtensible Host Controller Interface (xHCI) Controller.....	167

18.1.2 USB Dual Role Support - eXtensible Device Controller Interface (xDCI) Controller.....	167
18.1.3 AUX BIAS Control - USB Type-C Implementation with no Retimer.....	168
18.2 Signal Description.....	169
18.3 Integrated Pull-Ups and Pull-Downs.....	172
18.4 I/O Signal Planes and States.....	172
<b>19.0 PCI Express* (PCIe*).....</b>	<b>174</b>
19.1 Functional Description.....	174
19.1.1 Interrupt Generation.....	174
19.1.2 PCI Express* Power Management.....	175
19.1.3 Port 8xh Decode.....	176
19.1.4 Separate Reference Clock with Independent SSC (SRIS).....	176
19.1.5 Advanced Error Reporting.....	177
19.1.6 Single - Root I/O Virtualization (SR - IOV).....	177
19.1.7 SERR# Generation.....	177
19.1.8 Hot - Plug.....	178
19.1.9 PCI Express* Lane Polarity Inversion.....	178
19.1.10 Precision Time Measurement (PTM) .....	179
19.2 Signal Description.....	179
19.3 I/O Signal Planes and States.....	180
19.4 PCI Express* Port Support Feature Details.....	180
<b>20.0 Serial ATA (SATA).....</b>	<b>182</b>
20.1 Functional Description.....	182
20.1.1 SATA 6 Gb/s Support.....	182
20.1.2 SATA Feature Support.....	183
20.1.3 Hot - Plug Operation.....	183
20.1.4 Power Management Operation.....	183
20.1.5 SATA Device Presence.....	185
20.1.6 SATA LED.....	186
20.1.7 Advanced Host Controller Interface (AHCI) Operation.....	186
20.1.8 Enclosure Management (SGPIO Signals).....	187
20.2 Signals Description.....	189
20.3 Integrated Pull-Ups and Pull-Downs.....	190
20.4 I/O Signal Planes and States.....	190
<b>21.0 Universal Flash Storage (UFS).....</b>	<b>191</b>
21.1 Functional Description.....	191
21.2 Signals Description.....	191
21.3 I/O Signals Planes and States .....	191
<b>22.0 Graphics.....</b>	<b>193</b>
22.1 Processor Graphics.....	193
22.1.1 Media Support (Intel® QuickSync and Clear Video Technology HD).....	193
22.2 Platform Graphics Hardware Feature .....	196
22.2.1 Hybrid Graphics.....	196
<b>23.0 Display.....</b>	<b>197</b>
23.1 Display Technologies Support.....	197
23.2 Display Interfaces .....	197
23.2.1 Digital Display Interface (DDI) Signals.....	197

23.3 Display Configuration.....	197
23.4 Display Features.....	199
23.4.1 General Capabilities.....	199
23.4.2 Multiple Display Configurations.....	200
23.4.3 High-bandwidth Digital Content Protection (HDCP).....	200
23.4.4 DisplayPort*.....	200
23.4.5 High-Definition Multimedia Interface (HDMI*).....	202
23.4.6 embedded DisplayPort* (eDP*).....	204
23.4.7 MIPI* DSI.....	204
23.4.8 Integrated Audio.....	205
<b>24.0 High Precision Event Timer (HPET).....</b>	<b>206</b>
24.1 Feature Overview.....	206
24.1.1 Timer Accuracy.....	206
24.1.2 Timer Off-load.....	206
24.1.3 Interrupt Mapping.....	208
24.1.4 Periodic Versus Non-Periodic Modes.....	209
24.1.5 Enabling the Timers.....	211
24.1.6 Interrupt Levels.....	211
<b>25.0 8254 Timers.....</b>	<b>212</b>
25.1 Timer Programming.....	212
25.2 Reading from the Interval Timer.....	213
<b>26.0 Processor Sideband Signals.....</b>	<b>215</b>
26.1 Functional Description.....	215
26.2 Signal Description.....	215
26.3 Integrated Pull-Ups and Pull-Downs.....	215
26.4 I/O Signal Planes and States.....	215
<b>27.0 General Purpose Input and Output.....</b>	<b>217</b>
27.1 Functional Description.....	217
27.1.1 Configurable GPIO Voltage.....	218
27.1.2 GPIO Buffer Impedance Compensation.....	218
27.1.3 Interrupt / IRQ via GPIO Requirement.....	218
27.1.4 Programmable Hardware Debouncer.....	218
27.1.5 Integrated Pull-ups and Pull-downs.....	218
27.1.6 SCI / SMI# and NMI.....	219
27.1.7 Timed GPIO.....	219
27.1.8 GPIO Blink (BK) and Serial Blink (SBK).....	220
27.1.9 GPIO Ownership.....	220
27.1.10 Native Function and TERM Bit Setting.....	220
27.2 Signal Description.....	220
<b>28.0 GPIO Serial Expander.....</b>	<b>221</b>
28.1 Functional Description.....	221
28.2 Signal Description.....	222
28.3 Integrated Pull-ups and Pull-downs.....	222
<b>29.0 Intel® Serial I/O Inter-Integrated Circuit (I<sup>2</sup>C) Controllers.....</b>	<b>223</b>
29.1 Functional Description.....	224
29.1.1 Protocols Overview.....	224
29.1.2 DMA Controller.....	225



29.1.3	Reset.....	226
29.1.4	Power Management.....	226
29.1.5	Interrupts.....	227
29.1.6	Error Handling.....	227
29.1.7	I2C Setup/Hold Time.....	227
29.2	Signal Description.....	227
29.3	Integrated Pull-Ups and Pull-Downs.....	228
29.4	I/O Signal Planes and States.....	229
<b>30.0</b>	<b>Connectivity Integrated (CNVi).....</b>	<b>230</b>
30.1	Functional Description.....	230
30.2	Signal Description.....	231
30.3	Integrated Pull-ups and Pull-downs.....	233
30.4	I/O Signal Planes and States.....	233
<b>31.0</b>	<b>Integrated Sensor Hub (ISH).....</b>	<b>235</b>
31.1	Feature Overview.....	235
31.1.1	ISH I <sup>2</sup> C Controllers.....	236
31.1.2	ISH UART Controller.....	236
31.1.3	ISH GSPI Controller.....	236
31.1.4	ISH GPIOs.....	237
31.2	Functional Description.....	237
31.2.1	ISH Micro-Controller.....	237
31.2.2	SRAM.....	237
31.2.3	PCI Host Interface.....	237
31.2.4	Power Domains and Management.....	238
31.2.5	ISH IPC.....	238
31.2.6	ISH Interrupt Handling via IOAPIC (Interrupt Controller).....	238
31.3	Signal Description .....	239
31.4	Integrated Pull-Ups and Pull-Downs.....	239
31.5	I/O Signal Planes and States.....	240
<b>32.0</b>	<b>System Management.....</b>	<b>241</b>
32.1	Theory of Operation.....	241
32.1.1	Handling an Intruder.....	241
32.1.2	TCO Modes.....	242
<b>33.0</b>	<b>System Management Interface and SMLink.....</b>	<b>245</b>
33.1	Functional Description.....	245
33.2	Signal Description.....	245
33.3	Integrated Pull-Ups and Pull-Downs.....	246
33.4	I/O Signal Planes and States.....	246
<b>34.0</b>	<b>Host System Management Bus (SMBus) Controller.....</b>	<b>247</b>
34.1	Functional Description.....	247
34.1.1	Host Controller.....	247
34.1.2	SMBus Target Interface.....	254
34.2	SMBus Power Gating.....	261
34.3	Signal Description.....	261
34.4	Integrated Pull-Ups and Pull-Downs.....	261
34.5	I/O Signal Planes and States.....	262

<b>35.0 Serial Peripheral Interface (SPI)</b> .....	<b>263</b>
35.1 Functional Description.....	263
35.1.1 SPI0 for Flash.....	263
35.1.2 SPI0 support for TPM.....	267
35.2 Signal Description.....	268
35.3 Integrated Pull-Ups and Pull-Downs.....	268
35.4 I/O Signal Planes and States.....	268
35.5 VCCSPI Voltage (3.3 V or 1.8 V) Selection.....	269
<b>36.0 Enhanced Serial Peripheral Interface (eSPI)</b> .....	<b>270</b>
36.1 Functional Description.....	270
36.1.1 Operating Frequency.....	271
36.1.2 Protocols .....	271
36.1.3 WAIT States from eSPI Target.....	271
36.1.4 In-Band Link Reset.....	272
36.1.5 Target Discovery .....	272
36.1.6 Flash Sharing Mode.....	272
36.1.7 PECCI Over eSPI.....	272
36.1.8 Multiple OOB Initiator.....	272
36.1.9 Channels and Supported Transactions.....	272
36.2 Signal Description.....	278
36.3 Integrated Pull-Ups and Pull-Downs.....	278
36.4 I/O Signal Planes and States.....	279
<b>37.0 Intel® Serial IO Generic SPI (GSPI) Controllers</b> .....	<b>280</b>
37.1 Functional Description.....	280
37.1.1 Controller Overview.....	280
37.1.2 DMA Controller.....	281
37.1.3 Reset.....	282
37.1.4 Power Management.....	282
37.1.5 Interrupts.....	283
37.1.6 Error Handling.....	283
37.2 Signal Description.....	283
37.3 Integrated Pull-Ups and Pull-Downs.....	284
37.4 I/O Signal Planes and States.....	284
<b>38.0 Touch Host Controller (THC)</b> .....	<b>285</b>
38.1 Functional Description.....	285
38.2 Signal Description.....	286
38.3 Integrated Pull-Ups and Pull-Downs.....	287
38.4 I/O Signal Planes and States.....	287
<b>39.0 Intel® Serial I/O Universal Asynchronous Receiver/Transmitter (UART) Controllers</b> .....	<b>289</b>
39.1 Functional Description.....	290
39.1.1 UART Serial (RS-232) Protocols Overview.....	290
39.1.2 16550 8-bit Addressing - Debug Driver Compatibility.....	291
39.1.3 DMA Controller.....	291
39.1.4 Reset.....	292
39.1.5 Power Management .....	292
39.1.6 Interrupts.....	293

39.1.7 Error Handling.....	293
39.2 Signal Description.....	293
39.3 Integrated Pull-Ups and Pull-Downs.....	294
39.4 I/O Signal Planes and States.....	294
<b>40.0 Private Configuration Space Target Port ID.....</b>	<b>295</b>
<b>41.0 Testability.....</b>	<b>297</b>
41.1 JTAG.....	298
41.1.1 Testability Signals.....	298
41.1.2 Signal Description.....	299
41.1.3 I/O Signal Planes and States.....	299
41.2 Boundry Scan Sideband Signals.....	299
41.2.1 Signal Description.....	300
41.3 Intel® Trace Hub (Intel® TH).....	300
41.4 Direct Connect Interface (DCI).....	301
41.4.1 Out Of Band (OOB) Hosting DCI.....	301
41.4.2 USB 3.2 Hosting DCI.DBC.....	301
41.4.3 Platform Setup.....	302
<b>42.0 Digital Display Signals.....</b>	<b>303</b>
42.1 Signal Description.....	303
42.2 Embedded DisplayPort* (eDP*) Backlight Control Signals.....	304
42.3 Integrated Pull-Ups and Pull-Downs.....	305
42.4 I/O Signal Planes and States.....	305
<b>43.0 Miscellaneous Signals.....</b>	<b>306</b>
43.1 Signal Description.....	306
43.2 Reset and Miscellaneous Signals.....	307
43.3 Ground and Reserved Signals.....	308
43.4 Integrated Pull-Ups and Pull-Downs.....	309
43.5 I/O Signal Planes and States.....	309
43.6 Processor Internal Pull-Up / Pull-Down Terminations.....	309
<b>44.0 On Package Interface (OPI).....</b>	<b>310</b>
44.1 On Package Interface (OPI).....	310
44.1.1 OPI Support.....	310
44.1.2 Functional Description.....	310
<b>45.0 embedded Multi Media Card (eMMC*).....</b>	<b>311</b>
45.1 Features Supported.....	311
45.2 Functional Description.....	311
45.3 Signal Description.....	312
45.4 I/O Signal Planes and States.....	312

## Figures

1	Processor Line Platform Diagram.....	18
2	Flexible HSIO Lane Multiplexing .....	27
3	Example for PECI Host-Clients Connection.....	50
4	Example for PECI EC Connection.....	51
5	Device to Domain Mapping Structures .....	55
6	Processor Cache Hierarchy.....	62
7	Telemetry Aggregator.....	70
8	Processor Camera System.....	80
9	Processor Power States.....	84
10	Processor Package and IA Core C-States.....	85
11	Idle Power Management Breakdown of the Processor IA Cores.....	115
12	Package C-State Entry and Exit.....	118
13	Package Power Control.....	131
14	PROCHOT Demotion Signal Description .....	139
15	Integrated Clock Controller (ICC) Diagram.....	147
16	GPIO - Virtual Wire Index Bit Mapping .....	169
17	Generation of SERR# to Platform.....	177
18	Supported PCI Express* Link Configurations .....	180
19	Flow for Port Enable/Device Present Bits.....	186
20	Serial Data Transmitted over SGPIO Interface.....	189
21	Processor Display Architecture.....	199
22	DisplayPort* Overview.....	201
23	HDMI* Overview .....	203
24	MIPI* DSI Overview.....	205
25	Example of GSX Topology.....	221
26	Data Transfer on the I <sup>2</sup> C Bus.....	225
27	TCO Compatible Mode SMBus Configuration.....	242
28	Advanced TCO Mode.....	244
29	Flash Descriptor Regions.....	265
30	VCCSPI Voltage (3.3 V or 1.8 V) Selection.....	269
31	Basic eSPI Protocol.....	271
32	eSPI Target Request to PCH for PCH Temperature.....	275
33	PCH Response to eSPI Target with PCH Temperature .....	275
34	eSPI Target Request to PCH for PCH RTC Time.....	276
35	PCH Response to eSPI Target with RTC Time .....	277
36	THC Block Diagram.....	286
37	UART Serial Protocol .....	290
38	UART Receiver Serial Data Sample Points.....	291
39	Platform Setup with Intel® Trace Hub .....	300
40	Platform Setup with DCI Connection.....	302

## Tables

1	SKUs Supported.....	17
2	Terminology.....	23
3	Special Marks .....	26
4	CPUID Format.....	29
5	PCI Configuration Header.....	30
6	Host Device ID (DID0).....	30
7	Processor Graphics Device ID (DID2).....	30
8	Other Device ID.....	31
9	PCH Device and Revision ID .....	32
10	PCH ACPI Device ID for GPIO Controller.....	33
11	Package Mechanical Attributes.....	34
12	Fixed I/O Ranges Decoded by PCH.....	36
13	Variable I/O Decode Ranges .....	39
14	PCH Memory Decode Ranges (Processor Perspective).....	40
15	Boot Block Update Scheme.....	42
16	Pin Straps.....	44
17	Acronyms.....	71
18	Signal Descriptions.....	76
19	Integrated Pull-Ups and Pull-Downs.....	78
20	I/O Signal Planes and States.....	79
21	CSI-2 Lane Configuration.....	82
22	Acronyms.....	83
23	References.....	83
24	LPM_EN Register Mapping.....	91
25	General Power States for Systems Using the PCH.....	91
26	State Transition Rules for the PCH .....	92
27	System Power Plane.....	93
28	Causes of SMI and SCI .....	94
29	Sleep Types .....	97
30	Causes of Wake Events.....	98
31	Transitions Due to Power Failure .....	100
32	Supported Deep Sx Policy Configurations .....	101
33	Deep Sx Wake Events .....	102
34	Transitions Due to Power Button.....	103
35	Write Only Registers with Read Paths in ALT Access Mode.....	107
36	PIC Reserved Bits Return Values.....	107
37	SUSPWRDNACK/SUSWARN#/GPP_A13 Pin Behavior.....	110
38	SUSPWRDNACK During Reset.....	110
39	Causes of Host and Global Resets.....	112
40	System States .....	113
41	Integrated Memory Controller (IMC) States .....	114
42	G, S, and C Interface State Combinations .....	114
43	Core C-states .....	117
44	Package C-States.....	119
45	Deepest Package C-State Available.....	120
46	TCSS Power State .....	123
47	Power Rail Descriptions.....	125
48	Processor Ground Rails Signals .....	126
49	PCH Platform Voltage Rails.....	126
50	CORE_VID Signaling.....	127
51	VNN_CTRL Pin States.....	128
52	Assured Power (cTDP) Modes.....	133
53	Package Turbo Specifications .....	142
54	Junction Temperature Specifications .....	143

55	Error and Thermal Protection Signals.....	143
56	Signal Descriptions.....	147
57	I/O Signal Pin States.....	148
58	Acronyms.....	150
59	DDR4 Memory Interface.....	152
60	LP5 Memory Interface.....	153
61	DDR5 Memory Interface.....	154
62	DDR Support Matrix Table.....	154
63	DDR Technology Support Matrix.....	155
64	Supported DDR4 Non-ECC SoDIMM Module Configurations.....	155
65	Supported DDR5 Non-ECC SoDIMM Module Configurations.....	156
66	Supported DDR4 Memory Down Device Configurations.....	156
67	Supported DDR5 Memory Down Device Configurations.....	156
68	Supported LPDDR5 x32 DRAMs Configurations .....	156
69	Supported LPDDR5 x64 DRAMs Configurations.....	157
70	DDR System Memory Timing Support.....	157
71	LPDDR System Memory Timing Support .....	157
72	SA Speed Enhanced Speed Steps (SA-GV) and Gear Mode Frequencies .....	158
73	Ascending and Descending.....	160
74	USB-C* Port Configuration.....	164
75	USB-C* Lanes Configuration.....	165
76	Acronyms.....	167
77	References.....	167
78	Acronym.....	174
79	MSI Versus PCI IRQ Actions.....	174
80	Power Plane and States for PCI Express* Signals .....	180
81	PCI Express* Port Feature Details .....	180
82	Acronyms.....	182
83	References.....	182
84	Hardware Accelerated Video Decoding .....	194
85	Hardware Accelerated Video Encode .....	195
86	Display Ports Availability and Link Rate .....	197
87	Display Resolutions and Link Bandwidth for Multi-Stream Transport Calculations.....	201
88	DisplayPort Maximum Resolution.....	202
89	HDMI Maximum Resolution.....	204
90	Embedded DisplayPort Maximum Resolution.....	204
91	MIPI* DSI Maximum Resolution .....	205
92	Processor Supported Audio Formats over HDMI* and DisplayPort*.....	205
93	Legacy Replacement Routing.....	208
94	Counter Operating Modes.....	213
95	Acronyms.....	217
96	Native Function Signals Supporting Dynamic Termination Override.....	220
97	Acronyms.....	224
98	References.....	224
99	Acronyms.....	230
100	References.....	230
101	Acronyms.....	235
102	IPC Initiator -> Target Flows.....	238
103	Acronyms.....	241
104	Event Transitions that Cause Messages.....	243
105	Acronyms.....	245
106	Acronyms.....	247
107	References.....	247
108	I <sup>2</sup> C* Block Read.....	251
109	Enable for SMBALERT# .....	253

110	Enables for SMBus Target Write and SMBus Host Events.....	253
111	Enables for the Host Notify Command.....	253
112	Target Write Registers.....	255
113	Command Types.....	255
114	Target Read Cycle Format.....	256
115	Data Values for Target Read Registers.....	256
116	Host Notify Format.....	258
117	Target Read Cycle Format .....	259
118	Data Values for Target Read Registers.....	259
119	Enables for SMBus Target Write and SMBus Host Events.....	261
120	Acronyms.....	263
121	SPI0 Flash Regions.....	264
122	Region Size Versus Erase Granularity of Flash Components .....	265
123	Region Access Control Table.....	266
124	Acronyms.....	270
125	References.....	270
126	eSPI Channels and Supported Transactions.....	273
127	eSPI Virtual Wires (VW).....	273
128	Acronyms.....	280
129	Acronyms.....	285
130	Acronyms.....	290
131	Private Configuration Space Register Target Port IDs .....	295
132	Acronyms.....	298
133	References.....	298
134	Testability Signals.....	299
135	Power Planes and States for Testability Signals.....	299
136	BSSB Signals.....	300
137	Acronyms.....	303
138	Digital Display Signals.....	303
139	Embedded DisplayPort* (eDP*) Backlight Control Signals.....	304
140	Integrated Pull-Ups and Pull-Downs.....	305
141	I/O Signal Planes and States.....	305
142	Signal Descriptions.....	306
143	GND, RSVD, and NCTF Signals.....	308
144	Integrated Pull-Ups and Pull-Downs.....	309

## Revision History

---

Document Number	Revision Number	Description	Revision Date
759603	001	Initial Release	January 2023



## 1.0 Introduction

This document is intended for Original Equipment Manufacturers (OEMs), Original Design Manufacturers (ODM) and BIOS vendors creating products based on the Intel® Processor and Intel® Core™ i3 N-series.

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

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

Intel® Processor and Intel® Core™ i3 N-series processor is a 64-bit, multi-core processor built on 10-nanometer process technology.

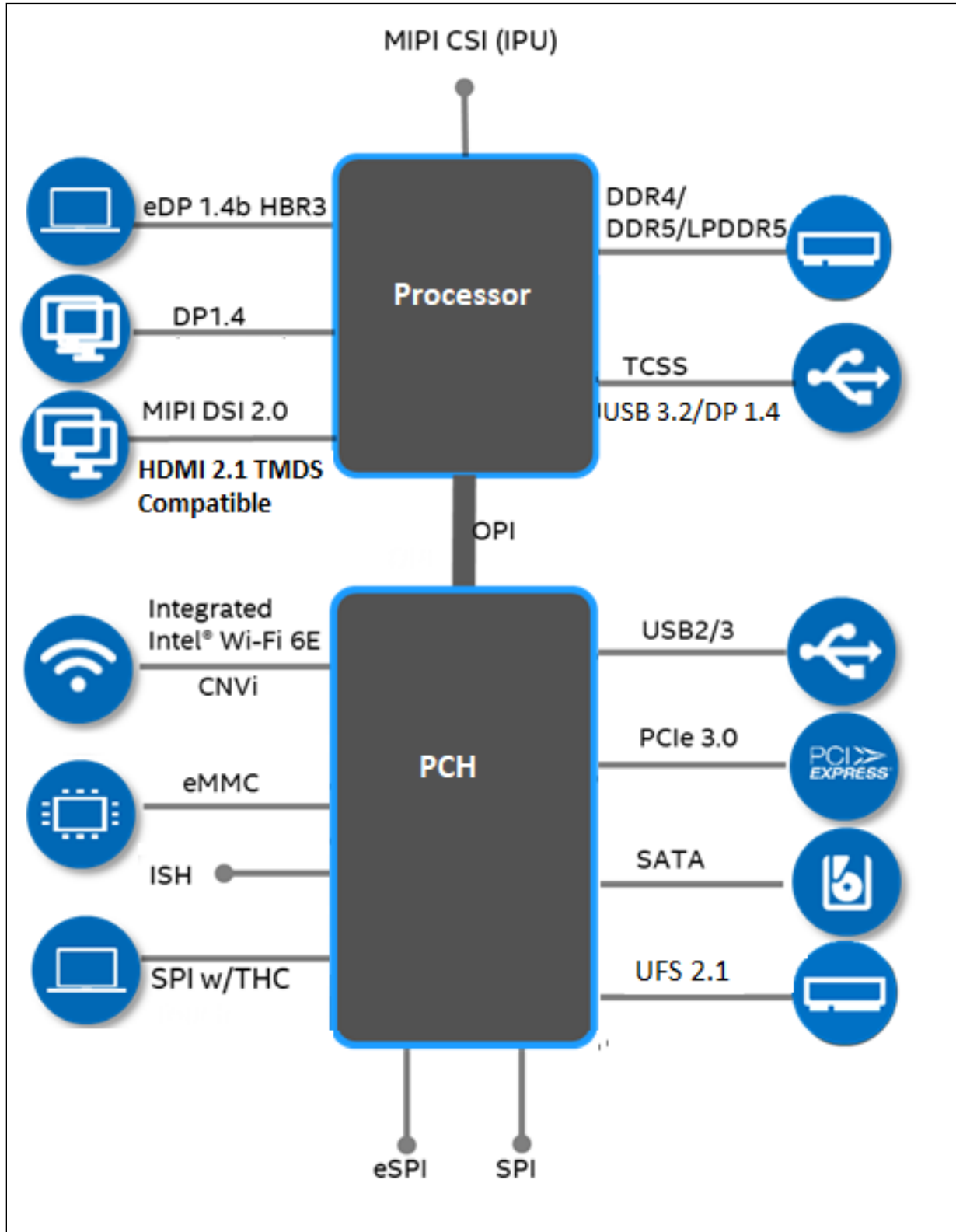
The N-Processor Line offered in a 1-Chip Platform that includes the Processor Die and N Platform Controller Hub (N PCH) die on the same package as the Multi-Chip Package (MCP).

The following table describes the different SKUs supported:

**Table 1. SKUs Supported**

Intel® Processor and Intel® Core™ i3 N-series	N305	N300	N200	N100
Proposed Branding	Intel® Core™ i3	Intel® Core™ i3	Intel® Processor	Intel® Processor
Use Condition	PC Client	PC Client	PC Client	PC Client
E-Cores	8	8	4	4
TDP (PL1)/PL2	15W/35W (cTDP 9W)	7W/25W	6W/25W	6W/25W
CPU HFM	1.8Ghz	0.8Ghz	1Ghz	0.8Ghz
CPU Max Burst Frequency	3.8Ghz	3.8Ghz	3.7Ghz	3.4Ghz
Gen 12LP	32EUs	32EUs	32EUs	24EUs
GFX HFM	1Ghz	0.6Ghz	0.450Ghz	0.400Ghz
GFX burst Frequency	1.25Ghz	1.25Ghz	0.75Ghz	0.75Ghz
TJ	0 to 105 °C	0 to 105 °C	0 to 105 °C	0 to 105 °C

Figure 1. Processor Line Platform Diagram



## 1.1 Processor Volatility Statement

Intel® Processor and Intel® Core™ i3 N-series processor families do not retain any end-user data when powered down and/or when the processor is physically removed.

---

**NOTE**

Powered down refers to the state which all processor power rails are off.

---

## 1.2 Package Support

The Intel® Processor and Intel® Core™ i3 N-series processor is available in the following packages:

BGA1264

- A 24 X 35 mm
- Substrate Z = 0.809+/-0.095 mm
- 1.386+/-0.109mm (Pre-SMT Z-height)

## 1.3 Supported Technologies

- PECCI – Platform Environmental Control Interface
- Intel® Virtualization Technology (Intel® VT-x)
- Intel® Virtualization Technology for Directed I/O (Intel® VT-d)
- Intel® APIC Virtualization Technology (Intel® APICv)
- Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI)
- PCLMULQDQ (Perform Carry-Less Multiplication Quad word) Instruction
- Intel® Secure Key
- Execute Disable Bit
- Intel® Boot Guard
- SMEP – Supervisor Mode Execution Protection
- SMAP – Supervisor Mode Access Protection
- SHA Extensions – Secure Hash Algorithm Extensions
- UMIP – User Mode Instruction Prevention
- RDPID – Read Processor ID
- Intel® Control-flow Enforcement Technology (Intel® CET)
- KeyLocker Technology
- Smart Cache Technology
- IA Core Level 1 and Level 2 Caches
- Intel® Turbo Boost Technology 2.0
- PAIR – Power Aware Interrupt Routing
- Intel SpeedStep® Technology

- Intel® Speed Shift Technology
- Intel® Advanced Vector Extensions 2 (Intel® AVX2)
- Intel® AVX2 Vector Neural Network Instructions (Intel® AVX2 VNNI)
- Intel® 64 Architecture x2APIC
- Intel® Dynamic Tuning Technology (Intel® DTT)
- Intel® GNA 3.0 (GMM and Neural Network Accelerator)
- Intel® Image Processing Unit (Intel® IPU)
- Cache Line Write Back (CLWB)
- Intel® Processor Trace
- Platform CrashLog
- Telemetry Aggregator
- Integrated Reference Clock PLL
- ACPI Power Management Logic Support, Revision 5.0a
- PCI Express Base Specification Revision 3.0
- Integrated Serial ATA Host controller 3.2, supports data transfer rates of up to 6 Gb/s on all ports
- USB 3.2 Gen 2x1 (10 Gb/s) eXtensible Host Controller (xHCI)
- USB 3.2 Gen 1x1 (5 Gb/s) Dual Role (eXtensible Device Controller - xDCI) Capability
- Serial Peripheral Interface (SPI)
- Enhanced Serial Peripheral Interface (eSPI)
- General Purpose Input Output (GPIO)
- Interrupt controller
- Timer functions
- System Management Bus (SMBus) Specification, Version 2.0
- Integrated Clock Controller (ICC)/Real Time Clock Controller (RTCC)
- Intel® High Definition Audio (Intel® HD Audio) and Intel® Smart Sound Technology (Intel® SST), supporting I<sup>2</sup>S, MIPI\* SoundWire\*, and DMIC.
- Intel® Serial I/O UART Host controllers
- Intel® Serial I/O I<sup>2</sup>C Host controllers
- Integrated Sensor Hub (ISH)
- Supports Intel® Virtualization Technology for Directed I/O (Intel® VT-d)
- JTAG Boundary Scan support
- Intel® Trace Hub (Intel® TH) and Direct Connect Interface (DCI) for debug
- Supports Intel® Converged Security Engine (Intel® CSE)
- Supports Integrated connectivity (CNVi)

### 1.3.1 API Support (Windows\*)

- Direct3D\* 2015, Direct3D 12, Direct3D 11.2, Direct3D 11.1, Direct3D 9, Direct3D 10, Direct2D
- OpenGL\* 4.5
- Open CL\* 2.1, Open CL 2.0, Open CL 1.2

DirectX\* extensions:

- PixelSync, Instant Access, Conservative Rasterization, Render Target Reads, Floating-point De-norms, Shared a Virtual memory, Floating Point atomics, MSAA sample-indexing, Fast Sampling (Coarse LOD), Quilted Textures, GPU Enqueue Kernels, GPU Signals processing unit. Other enhancements include color compression.

Gen 12 architecture delivers hardware acceleration of Direct X\* 12 Render pipeline comprising of the following stages: Vertex Fetch, Vertex Shader, Hull Shader, Tessellation, Domain Shader, Geometry Shader, Rasterizer, Pixel Shader, Pixel Output.

## 1.4 Power Management Support

### 1.4.1 Processor Core Power Management

- Full support of ACPI C-states as implemented by the following processor C-states:
  - C0, C1, C1E, C6, C8, C10
- Enhanced Intel SpeedStep® Technology
- Intel® Speed Shift Technology

Refer to [Processor IA Core Power Management](#) on page 114 for more information.

### 1.4.2 System Power Management

- S0/S0ix, S3, S4, S5

Refer to [Power Management](#) on page 83 for more information.

### 1.4.3 Memory Controller Power Management

- Disabling Unused System Memory Outputs
- DRAM Power Management and Initialization
- Initialization Role of CKE
- Conditional Self-Refresh
- Dynamic Power Down
- DRAM I/O Power Management
- DDR Electrical Power Gating (EPG)
- Power Training

Refer to [Integrated Memory Controller \(IMC\) Power Management](#) on page 160 for more information.

## 1.4.4 Processor Graphics Power Management

### Memory Power Savings Technologies

- Intel® Rapid Memory Power Management (Intel® RMPM)
- Intel® Smart 2D Display Technology (Intel® S2DDT)

### Display Power Savings Technologies

- Intel® (Seamless and Static) Display Refresh Rate Switching (DRRS) with eDP\* port
- Intel® Automatic Display Brightness
- Smooth Brightness
- Intel® Display Power Saving Technology (Intel® DPST 6.3)
- Panel Self-Refresh 2 (PSR 2)
- Low Power Single Pipe (LPSP)

### Graphics Core Power Savings Technologies


- Graphics Dynamic Frequency
- Intel® Graphics Render Standby Technology (Intel® GRST)
- Dynamic FPS (DFPS)

## 1.5 Thermal Management Support

- Digital Thermal Sensor
- Intel® Adaptive Thermal Monitor
- THERMTRIP# and PROCHOT# support
- On-Demand Mode
- Memory Open and Closed Loop Throttling
- Memory Thermal Throttling
- External Thermal Sensor (TS-on-DIMM and TS-on-Board)
- Render Thermal Throttling
- Fan Speed Control with DTS
- Intel® Turbo Boost Technology 2.0 Power Control
- Intel® Dynamic Tuning Technology (Intel® DTT)

Refer to [Thermal Management](#) on page 129 for more information.

## 1.6 Ball-out Information

For information on the N processor ball information, download the pdf, click  on the navigation pane and refer the spreadsheet, **759603-001\_Ballout.xlsx**.

## 1.7 Processor Testability

A DCI on-board connector should be placed, to enable the full debug capabilities. For Intel® Processor and Intel® Core™ i3 N-series processor SKUs, a Direct Connect Interface Tool connector is highly recommended to enable lower C-state to debug.

The processor includes boundary-scan for board and system level testability. Refer to the appropriate Processor Testability Information - Boundary Scan Description Language (BSDL) file.

## 1.8 Operating Systems Support

Windows* 11 (64-bit only) - 22'H2, Windows* 10	Android*	Linux*	Chrome* OS
Yes	No	Yes	Yes

## 1.9 Terminology and Special Marks

### Terminology Usage

This document uses the term **initiator** and **target** (formerly known as **master** and **slave**).

**Table 2. Terminology**

Term	Description
4K	Ultra High Definition (UHD)
AES	Advanced Encryption Standard
AGC	Adaptive Gain Control
API	Application Programming Interface
AVC	Advanced Video Coding
BLT	Block Level Transfer
BPP	Bits per Pixel
CDR	Clock and Data Recovery
CTLE	Continuous Time Linear Equalizer
eDP*	embedded Display Port*
DDC	Digital Display Channel
DDI	Digital Display Interface
DSI	Display Serial Interface
DDR4	Fourth-Generation Double Data Rate SDRAM Memory Technology
DDR5	Fifth-Generation Double Data Rate SDRAM Memory Technology
DFE	Decision Feedback Equalizer
DMA	Direct Memory Access
DPPM	Dynamic Power Performance Management
<i>continued...</i>	

Term	Description
DMI	Direct Media Interface
DP*	DisplayPort*
DSC	Display Stream Compression
DSI	Display Serial Interface
DTS	Digital Thermal Sensor
ECC	Error Correction Code - used to fix DDR transactions errors
eDP*	Embedded DisplayPort*
EU	Execution Unit in the Graphics Processor
FIA	Flex IO Adapter
FIVR	Fully Integrated Voltage Regulator
GSA	Graphics in System Agent
GNA	Gauss Newton Algorithm
HDCP	High-Bandwidth Digital Content Protection
HDMI*	High Definition Multimedia Interface
IMC	Integrated Memory Controller
Intel® 64 Technology	64-bit memory extensions to the IA-32 architecture
Intel® DPST	Intel® Display Power Saving Technology
Intel® PTT	Intel® Platform Trust Technology
Intel® VT	Intel® Virtualization Technology. Processor Virtualization, when used in conjunction with Virtual Machine Monitor software, enables multiple, robust independent software environments inside a single platform.
Intel® VT-d	Intel® Virtualization Technology (Intel® VT) for Directed I/O. Intel® VT-d is a hardware assist, under system software (Virtual Machine Manager or OS) control, for enabling I/O device Virtualization. Intel® VT-d also brings robust security by providing protection from errant DMAs by using DMA remapping, a key feature of Intel® VT-d.
ITH	Intel® Trace Hub
IOV	I/O Virtualization
IPU	Image Processing Unit
LFM	Low Frequency Mode. corresponding to the Enhanced Intel SpeedStep® Technology's lowest voltage/frequency pair. It can be read at MSR CEh [47:40]. For more information, refer to appropriate BIOS Specification.
LLC	Last Level Cache
LPDDR5	Low Power Double Data Rate SDRAM memory technology additional power save.
LPSP	Low-Power Single Pipe
LSF	Lowest Supported Frequency. This frequency is the lowest frequency where manufacturing confirms logical functionality under the set of operating conditions.
LTR	The Latency Tolerance Reporting (LTR) mechanism enables Endpoints to report their service latency requirements for Memory Reads and Writes to the Root Complex, so that power management policies for central platform resources (such as main memory, RC internal interconnects, and snoop resources) can be implemented to consider Endpoint service requirements.
<b>continued...</b>	



Term	Description
MCP	Multi-Chip Package - includes the processor and the PCH.
MFM	Minimum Frequency Mode. MFM is the minimum ratio supported by the processor and can be read from MSR CEh [55:48]. For more information, refer to the appropriate BIOS specification.
MIPI-DSI	Mobile Industry Processor Interface Display Serial Interface
MLC	Mid-Level Cache
MPEG	Motion Picture Expert Group, international standard body JTC1/SC29/WG11 under ISO/IEC that has defined audio and video compression standards such as MPEG-1, MPEG-2, and MPEG-4, etc.
NCTF	Non-Critical to Function. NCTF locations are typically redundant ground or non-critical reserved balls/lands, so the loss of the solder joint continuity at end of life conditions will not affect the overall product functionality.
PCH	Platform Controller Hub. The chipset with centralized platform capabilities including the main I/O interfaces along with display connectivity, audio features, power management, manageability, security, and storage features. The PCH may also be referred to as "chipset".
PECI	Platform Environment Control Interface
PL1, PL2 (a.k.a Maximum Turbo power), PL3	Power Limit 1, Power Limit 2, Power Limit 3
PMIC	Power Management Integrated Circuit
Processor	The 64-bit multi-core component (package)
Processor Core	The term "processor core" refers to the Silicon die itself, which can contain multiple execution cores. Each execution core has an instruction cache, data cache, and 256-KB L2 cache. All execution cores share the LLC.
Processor Graphics	Intel® Processor Graphics
PSR	Panel Self-Refresh
PSx	Power Save States (PS0, PS1, PS2, PS3, PS4)
Rank	A unit of DRAM corresponding to four to eight devices in parallel, ignoring ECC. These devices are usually, but not always, mounted on a single side of a SoDIMM.
SCI	System Control Interrupt. SCI is used in the ACPI protocol.
SHA	Secure Hash Algorithm
SSC	Spread Spectrum Clock
SSIC	SuperSpeed Inter-Chip
Storage Conditions	Refer <a href="#">Package Storage Specifications</a> on page 35.
STR	Suspend to RAM
TAC	Thermal Averaging Constant
TCC	Thermal Control Circuit
Processor Base Power (a.k.a TDP)	Thermal Design Power
TTV Processor Base Power (a.k.a TDP)	Thermal Test Vehicle TDP
V <sub>CC</sub>	Processor Core Power Supply

continued...

Term	Description
V <sub>CCGT</sub>	Processor Graphics Power Supply
V <sub>CCSA</sub>	System Agent Power Supply
VLD	Variable Length Decoding
VPID	Virtual Processor ID
V <sub>SS</sub>	Processor Ground
D0ix-states	USB controller power states ranging from D0i0 to D0i3, where D0i0 is fully powered on and D0i3 is primarily powered off. Controlled by SW.
S0ix-states	Processor residency idle standby power states.

**Table 3. Special Marks**

Mark	Definition
[ ]	Brackets ([ ]) sometimes follow a ball, pin, registers or a bit name. These brackets enclose a range of numbers, for example, TCP[2:0]_TXRX_P[1:0] may refer to four USB-C* pins or EAX[7:0] may indicate a range that is 8 bits length.
_N / # / B	A suffix of _N or # or B indicates an active low signal. For example, CATERR# _N does not refer to a differential pair of signals such as CLK_P, CLK_N
0x000	Hexadecimal numbers are identified with an x in the number. All numbers are decimal (base 10) unless otherwise specified. Non-obvious binary numbers have the 'b' enclosed at the end of the number. For example, 0101b

## 1.10 Flexible High Speed I/O

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

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

The figure below shows the High Speed I/O (HSIO) lane multiplexing:

Figure 2. Flexible HSIO Lane Multiplexing

PCH-N	Flex HSIO Lanes										
	0	1	2	3	6	8	9	10	11		
HSIO Type and Lane	USB 3.2 Gen 1x1/2x1 #1	USB 3.2 Gen 1x1/2x1 #2	USB 3.2 Gen 1x1/2x1 #3	USB 3.2 Gen 1x1/2x1 #4	PCIe #7	PCIe #9	PCIe #10	PCIe #11	PCIe #12		
	PCIe #1	PCIe #2	PCIe #3	PCIe #4		UFS 1x2 Lane 0    Lane 1		SATA 0	SATA 1		

The 9 Flexible HSIO Lanes [11:8,6,3:0] supports the following configurations:

- Up to nine PCIe\* Lanes
  - A maximum of five PCIe\* Root Ports (or devices) can be enabled
  - PCIe\* Lanes 1-4 (PCIe\* Controller #1), 7 (PCIe\* Controller #2), and 9-12 (PCIe\* Controller #3) must be individually configured.
- Up to four USB 3.2 Gen 1x1/2x1 Lanes
  - A maximum of four USB 3.2 Gen 1x1/2x1 Ports (or devices) can be enabled.
  - USB 3.2 Gen 1x1 = 5 GT/s
  - USB 3.2 Gen 2x1 = 10 GT/s
- Up to two SATA Lanes
  - A maximum of two SATA Ports (or devices) can be enabled.
- Supports two Lane 1x2 Universal Flash Storage (UFS)
- For unused USB 3.2/PCIe\* Combo Lanes, the unused lanes must be statically assigned to PCIe\* or USB 3.2 via the USB 3.2/PCIe\* Combo Port Soft Straps through the Intel Flash Image Tool (FIT) tool.

### 1.10.1 Flexible I/O Lane Selection

HSIO lane configuration and type is statically selected by soft straps, which are managed through the Platform Flash Image Tool, available as part of Intel® CSE releases.

---

**NOTE**

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

---

**PCIe\*/SATA Lane Selection**

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

**1.11 Related Documents**

Document	Document Number
Intel® Processor and Intel® Core™ i3 N-series Datasheet Volume 2 of 2	759604

## 2.0 Processor and PCH Device IDs

### 2.1 CPUID

**Table 4. CPUID Format**

SKU	CPUID	Reserved [31:28]	Extended Family [27:20]	Extended Model [19:16]	Reserved [15:14]	Processor Type [13:12]	Family Code [11:8]	Model Number [7:4]	Stepping ID [3:0]
Intel® Processor and Intel® Core™ i3 N-series +N0	B06E0h	Reserved	0000000b	1011b	Reserved	00b	0110b	1110b	0000b

- The Extended Family, Bits [27:20] are used in conjunction with the Family Code, specified in Bits [11:8], to indicate whether the processor belongs to Intel® Core™ processor family.
- The Extended Model, Bits [19:16] in conjunction with the Model Number, specified in Bits [7:4], are used to identify the model of the processor within the processor's family.
- The Family Code corresponds to Bits [11:8] of the EDX register after RESET, Bits [11:8] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the generation field of the Device ID register accessible through Boundary Scan.
- The Model Number corresponds to Bits [7:4] of the EDX register after RESET, Bits [7:4] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the model field of the Device ID register accessible through Boundary Scan.
- The Stepping ID in Bits [3:0] indicates the revision number of that model.
- When EAX is initialized to a value of '1', the CPUID instruction returns the Extended Family, Extended Model, Processor Type, Family Code, Model Number and Stepping ID value in the EAX register. Note that the EDX processor signature value after reset is equivalent to the processor signature output value in the EAX register.

Cache and TLB descriptor parameters are provided in the EAX, EBX, ECX and EDX registers after the CPUID instruction is executed with a 2 in the EAX register.

### 2.2 PCI Configuration Header

Every PCI-compatible function has a standard PCI configuration header, as shown in the table below. This includes mandatory registers (Bold) to determine which driver to load for the device. Some of these registers define ID values for the PCI function, which are described in this chapter.

**Table 5. PCI Configuration Header**

Byte3	Byte2	Byte1	Byte0	Address
<b>Device ID</b>		<b>Vendor ID (0x8086)</b>		00h
Status		Command		04h
Class Code			<b>Revision ID</b>	08h
BIST	Header Type	Latency Timer	Cache Line Size	0Ch
Base Address Register0 (BAR0)				10h
Base Address Register1 (BAR1)				14h
Base Address Register2 (BAR2)				18h
Base Address Register3 (BAR3)				1Ch
Base Address Register4 (BAR4)				20h
Base Address Register5 (BAR5)				24h
Card-bus CIS Pointer				28h
<b>Subsystem ID</b>		<b>Subsystem Vendor ID</b>		2Ch
Expansion ROM Base Address				30h
Reserved			Capabilities Pointer	34h
Reserved				38h
Maximum Latency	Minimum Grant	Interrupt Pin	Interrupt Line	3Ch

## 2.3 Processor Device IDs

This section specifies the device IDs of the processor.

**Table 6. Host Device ID (DID0)**

Platform	Device ID
0+8	4617h
0+4 (Intel® Processor N200)	461Bh
0+4 (Intel® Processor N100)	461Ch

**Table 7. Processor Graphics Device ID (DID2)**

Platform	Processor Step	GT SKU	Device ID
0+8	N0	32EU	46D0h
0+4	N0	32EU	46D0h
0+4	N0	24EU	46D1h

**Table 8. Other Device ID**

Device	Bus / Device / Function	DID
Dynamic Tuning Technology (DTT)	0 / 4 / 0	461Dh
IPU(IMGU)	0 / 5 / 0	462Eh
Gauss Newton Algorithm (GNA)	0 / 8 / 0	467Eh
Intel® Trace Hub	0 / 9 / 0	466Fh
Crash Log and Telemetry	0 / 10 / 0	467Dh
USB xHCI	0 / 13 / 0	464Eh
USB xDCI	0 / 13 / 1	465Eh

## 2.4 PCH Device and Revision IDs

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

The RID register reports one of the two possible values:

- Stepping Revision Identification (SRID)
- Compatible Revision ID (CRID)

The default power-on value for the RID register is SRID. The assigned value is based on the product's stepping. CRID is intended for the corporate Intel® Stable Image Platform Program (Intel® SIPP). CRID is normally identical to the SRID value of a previous production stepping of the product with which the new stepping is deemed compatible. Intel® SIPP allows an OS image built on the earlier stepping to be used on any new compatible stepping(s). Three CRID values are possible and can be used to manage software images.

---

### NOTE

SRID and CRID are not addressable PCI registers. The SRID and CRID value are reflected through the RID register when appropriately selected.

---

Following reset, the SRID value can be read from the RID registers of all PCH devices and functions.

To select either SRID or CRID to be reflected in the RID registers:

1. BIOS needs to write appropriate value into the Configured Revision ID (CRID) register located in the PMC MMIO space. Refer to Intel® Processor and Intel® Core™ i3 N-series Datasheet Volume 2 of 2 ([#759604](#)) for definition details of the register.
2. BIOS must write this register with the appropriate value after S3/S4/S5 states and after PLTRST# events.

After CRID is selected and applied by BIOS, software will not be able to obtain the original SRID value of the PCH by reading the PCH RID registers. Customers implementing CRID who also want to determine the SRID in runtime may develop their own tool. For example, BIOS can capture the SRID value before BIOS applies

CRID and store that value in a runtime accessible place (that is, SMBIOS, ACPI Type 4 Memory, NVRAM, CMOS) so that it can be read by the customer tool later. Alternatively, the BIOS can store the SRID value and display this information in BIOS setup while reporting that CRID is enabled.

BIOS needs to check CRID\_UIP bit (in PMC MMIO space) as a part of the update flow. PMC HW sets this bit to indicate that SetID broadcast flow has been requested by BIOS. This bit is cleared by PMC FW only when the completion/s of SetIDVal message is received by PMC. BIOS is required to read this bit as cleared before writing to the CRID register (to request a CRID update). BIOS is also required to poll on reads to this bit until it detects the bit as cleared after BIOS has written to the CRID register.

**Table 9. PCH Device and Revision ID**

Dev ID	Device Function - Device Description	Note
5480 - 549F	D31:F0 - eSPI Controller	<b>PCH Device ID :</b> 5481
54A0	D31:F1 - P2SB	
54A1	D31:F2 - PMC	
54A3	D31:F4 - SMBus	
54A4	D31:F5 - SPI (flash) Controller	
54A6	D31:F7 - Intel® Trace Hub (Intel® TH)	
54A8	D30:F0 - UART #0	
54A9	D30:F1 - UART #1	
54AA	D30:F2 - GSPI #0	
54AB	D30:F3 - GSPI #1	
54B0	D29:F0 - PCI Express* Root Port #9	
54B1	D29:F1 - PCI Express* Root Port #10	
54B2	D29:F2 - PCI Express* Root Port #11	
54B3	D29:F3 - PCI Express* Root Port #12	
54B8	D28:F0 - PCI Express* Root Port #1	
54B9	D28:F1 - PCI Express* Root Port #2	
54BA	D28:F2 - PCI Express* Root Port #3	
54BB	D28:F3 - PCI Express* Root Port #4	
54BE	D28:F6 - PCI Express* Root Port #7	
54C4	D26:F0 - SCS	embedded Multi Media Card (eMMC) Controller
54C5	D25:F0 - I <sup>2</sup> C Controller #4	
54C6	D25:F1 - I <sup>2</sup> C Controller #5	
54C7	D25:F2 - UART #2	
54C8 - 54CF	D31:F3 - Intel® High Definition Audio (Intel®HD Audio) (Audio, Voice, Speech)	
54D0	D16:F6 - Touch Host Controller #0 (THC #0)	
<i>continued...</i>		



Dev ID	Device Function - Device Description	Note
54D1	D16:F7 - Touch Host Controller #1 (THC #1)	
54D3	D23:F0 - SATA Controller (AHCI)	
54DA	D17:F0 - UART Controller #3	
54E0	D22:F0 - Intel® CSE: HECI #1	
54E1	D22:F1 - Intel® CSE: HECI #2	
54E4	D22:F4 - Intel® CSE: HECI #3	
54E5	D22:F5 - Intel® CSE: HECI #4	
54E8	D21:F0 - I <sup>2</sup> C Controller #0	
54E9	D21:F1 - I <sup>2</sup> C Controller #1	
54EA	D21:F2 - I <sup>2</sup> C Controller #2	
54EB	D21:F3 - I <sup>2</sup> C Controller #3	
54ED	D20:F0 - USB 3.2 Gen 2x1 (10 Gb/s) xHCI HC	
54EE	D20:F1 - USB 3.2 Gen 1x1 (5 Gb/s) Device Controller (xDCI)	
54EF	D20:F2 - Shared SRAM	
54F0 - 54F3	D20:F3 - CNVi: Wi-Fi*	
54FB	D18:F6 - GSPI #2	
54FF	D18:F7 - SCS	Universal Flash Storage(UFS)
54FC	D18:F0	Integrated Sensor Hub

**Table 10. PCH ACPI Device ID for GPIO Controller**

ACPI ID	Note
INTC1057	

## 3.0 Package Mechanical Specifications

### 3.1 Package Mechanical Attributes

The Intel® Processor and Intel® Core™ i3 N-series Processor Lines use a Flip Chip technology available in a Ball Grid Array (BGA) package. The following table provides an overview of the package mechanical attributes. For specific dimensions (die size, die location, and so on), refer to the processor package mechanical drawings.

**Table 11. Package Mechanical Attributes**

Package	Parameter	Processor Line
Package Technology	Package Type	Flip Chip Ball Grid Array
	Interconnect	Ball Grid Array (BGA)
	Lead Free	Yes
	Halogenated Flame Retardant Free	Yes
Package Configuration	Solder Ball Composition	SAC405
	Ball/Pin Count	1264
	Grid Array Pattern	Balls anywhere
	Land Side Capacitors	Yes
	Die Side Capacitors	No
	Die Configuration	2 Dice Multi Chip package (MCP)
Package Dimensions	Nominal Package Size	24 x 35 mm <sup>2</sup>
	Z	Substrate Z = 0.809+/-0.095 mm 1.386+/-0.109mm (Pre-SMT Z-height)
	Minimum Ball/Pin pitch	0.62 mm BP

### 3.2 Package Loading and Die Pressure Specifications

Intel has defined the maximum total compressive load limits that can be applied to the package. These values should not be exceeded by the system design.

#### 3.2.1 Die Pressure Specifications

A more relevant metric for concentrated loading is chosen by Intel based on the physics of failure to evaluate die damage risk due to thermal solution enabling

Static Compressive pressure refers to the long term steady state pressure applied to the die from the thermal solution after system assembly is complete.

Transient Compressive pressure refers to the pressure on the dice at any moment during the thermal solution assembly/disassembly procedures. Other system procedures such as repair/rework can also cause high pressure loading to occur on the die and should be evaluated to ensure these limits are not exceeded.

**Metric:** This metric is pressure over a 2 mm x 2 mm area

### 3.3 Package Storage Specifications

Parameter	Description	Minimum	Maximum	Notes
T <sub>ABSOLUTE STORAGE</sub>	The non-operating device storage temperature. Damage (latent or otherwise) may occur when subjected to this temperature for any length of time in Intel Original sealed moisture barrier bag and / or box.	-25°C	125°C	
T <sub>SUSTAINED STORAGE</sub>	The ambient storage temperature limit (in shipping media) for the sustained period of time	-5°C	40°C	
RH <sub>SUSTAINED STORAGE</sub>	The maximum device storage relative humidity for the sustained period of time as specified below in Intel Original sealed moisture barrier bag and / or box	60% @ 24°C		
TIME <sub>SUSTAINED STORAGE</sub>	Maximum time: associated with customer shelf life in Intel Original sealed moisture barrier bag and / or box	NA	<b>Moisture Sensitive Devices:</b> 60 months from bag seal date; Non-moisture sensitive devices: 60 months from lot date	
Storage Conditions	Processors in a non-operational state may be installed in a platform, in a tray, boxed, or loose and may be sealed in airtight package or exposed to free air. Under these conditions, processor landings should not be connected to any supply voltages, have any I/Os biased, or receive any clocks. Upon exposure to "free air" (that is, unsealed packaging or a device removed from packaging material), the processor should be handled in accordance with moisture sensitivity labeling (MSL) as indicated on the packaging material. Boxed Land Grid Array packaged (LGA) processors are MSL 1 ('unlimited' or unaffected) as they are not heated in order to be inserted in the socket.			
<p><i>Notes:</i> 1. T<sub>ABSOLUTE STORAGE</sub> applies to the un-assembled component only and does not apply to the shipping media, moisture barrier bags or desiccant. Refers to a component device that is not assembled in a board or socket that is not to be electrically connected to a voltage reference or I/O signals.</p> <p>2. Specified temperatures are based on data collected. Exceptions for surface mount re-flow are specified by applicable JEDEC J-STD-020 documents. The JEDEC, J-STD-020 moisture level rating and associated handling practices apply to all moisture sensitive devices removed from the moisture barrier bag.</p> <p>3. Post board attaches storage temperature limits are not specified for non-Intel branded boards. Consult your board manufacturer for storage specifications.</p>				

## 4.0 Memory Mapping

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

### 4.1 Functional Description

This section provides information on the following topics:

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

#### 4.1.1 PCI Devices and Functions

The PCH incorporates a variety of PCI devices and functions, as shown in the following table. If for some reason, the particular system platform does not want to support any one of the Device Functions, with the exception of D30:F0, they can individually be disabled. When a function is disabled, it does not appear to the software. A disabled function will not respond to any register reads or writes, ensuring that these devices appear hidden to software.

#### 4.1.2 Fixed I/O Address Ranges

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

**NOTE**

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

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

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

**Table 12. Fixed I/O Ranges Decoded by PCH**

I/O Address	Read Target	Write Target	Internal Unit (unless[E]: External) <sup>2</sup>	Separate Enable/Disable
20h – 21h	Interrupt Controller	Interrupt Controller	Interrupt	None
24h – 25h	Interrupt Controller	Interrupt Controller	Interrupt	None
<i>continued...</i>				

I/O Address	Read Target	Write Target	Internal Unit (unless[E]: External) <sup>2</sup>	Separate Enable/Disable
28h – 29h	Interrupt Controller	Interrupt Controller	Interrupt	None
2Ch – 2Dh	Interrupt Controller	Interrupt Controller	Interrupt	None
2E-2F	Super I/O	Super I/O	[E] Forwarded to eSPI	Yes. ESPI_IOD_IOE.SE
30h – 31h	Interrupt Controller	Interrupt Controller	Interrupt	None
34h – 35h	Interrupt Controller	Interrupt Controller	Interrupt	None
38h – 39h	Interrupt Controller	Interrupt Controller	Interrupt	None
3Ch – 3Dh	Interrupt Controller	Interrupt Controller	Interrupt	None
40h	Timer/Counter	Timer/Counter	8254 Timer	None
42h-43h	Timer/Counter	Timer/Counter	8254 Timer	None
4E-4F	Microcontroller	Microcontroller	[E] Forwarded to eSPI	Yes. ESPI_IOD_IOE.ME2
50h	Timer/Counter	Timer/Counter	8254 Timer	None
52h-53h	Timer/Counter	Timer/Counter	8254 Timer	None
60h	Keyboard Controller	Keyboard Controller	[E] Forwarded to eSPI	Yes, with 64h. ESPI_IOD_IOE.KE
61h	NMI Controller	NMI Controller	CPU I/F	None
62h	Microcontroller	Microcontroller	[E] Forwarded to eSPI	Yes, with 66h. ESPI_IOD_IOE.ME1
63h	NMI Controller <sup>1</sup>	NMI Controller <sup>1</sup>	CPU I/F	Yes, alias to 61h. GIC.P61AE
64h	Keyboard Controller	Keyboard Controller	[E] Forwarded to eSPI	Yes, with 60h. ESPI_IOD_IOE.KE
65h	NMI Controller <sup>1</sup>	NMI Controller <sup>1</sup>	CPU I/F	Yes, alias to 61h. GIC.P61AE
66h	Microcontroller	Microcontroller	[E] Forwarded to eSPI	Yes, with 62h. ESPI_IOD_IOE.ME1
67h	NMI Controller <sup>1</sup>	NMI Controller <sup>1</sup>	CPU I/F	Yes, alias to 61h. GIC.P61AE
70h	RTC Controller	NMI and RTC Controller	RTC	None
71h	RTC Controller	RTC Controller	RTC	None
72h	RTC Controller	RTC Controller	RTC	None. Alias to 70h if RC.UE <sup>4</sup> =0, else 72h
73h	RTC Controller	RTC Controller	RTC	None. Alias to 71h if RC.UE='0', else 73h
74h	RTC Controller	RTC Controller	RTC	None
75h	RTC Controller	RTC Controller	RTC	None

*continued...*

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

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

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

### 4.1.3 Variable I/O Decode Ranges

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

#### WARNING

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

**Table 13. Variable I/O Decode Ranges**

Range Name <sup>1</sup>	Mappable	Size (Bytes)	Target
ACPI	Anywhere in 64K I/O Space	256	Power Management
SMBus	Anywhere in 64K I/O Space	32	SMB Unit
TCO	Anywhere in 64K I/O Space	32	SMB Unit
Parallel Port	3 ranges in 64K I/O Space	8	eSPI
Serial Port 1	8 Ranges in 64K I/O Space	8	eSPI
Serial Port 2	8 Ranges in 64K I/O Space	8	eSPI
Serial Port 3	8 Ranges in 64K I/O space	8	eSPI
Floppy Disk Controller	2 Ranges in 64K I/O Space	8	eSPI
IO Trapping Ranges	Anywhere in 64K I/O Space	1 to 256 Bytes	Trap
Serial ATA Index/Data Pair	Anywhere in 64K I/O Space	16	SATA Host Controller
PCI Express* Root Ports	Anywhere in 64K I/O Space	I/O Base/Limit	PCIe port N (N=1 to 4, 7, 9 to 12)

Note: All ranges are decoded directly from OPI.

## 4.2 Memory Map

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

PCIe cycles generated by external PCIe initiators will be positively decoded unless they fall in the PCI-PCI bridge memory forwarding ranges (those addresses are reserved for PCI peer-to-peer traffic). Software must not attempt locks to the PCH’s memory-mapped I/O ranges.

**Table 14. PCH Memory Decode Ranges (Processor Perspective)**

Memory Range	Target	Dependency/Comments
000E 0000 - 000E FFFF	eSPI or SPI	Bit 6 in BIOS Decode Enable Register is set
000F 0000 - 000F FFFF	eSPI or SPI	Bit 7 in BIOS Decode Enable Register is set
FECX X000 - FECX X040	I/O(x)APIC inside PCH	XX controlled via APIC Range Select (ASEL) field and APIC Enable (AEN) bit
FECX X000 - FECX XFFF	PCIe port N (N=1 to 4, 7, 9 to 12)	X controlled via PCIe root port N IOxAPIC Range Base/Limit registers and Port N I/OxApic Enable (PAE) is set
FEC1 0000 - FEC1 7FFF	PCIe port 1	PCIe root port 1 I/OxApic Enable (PAE) is set
FEC1 8000 - FEC1 FFFF	PCIe port 2	PCIe root port 2 I/OxApic Enable (PAE) is set
FEC2 0000 - FEC2 7FFF	PCIe port 3	PCIe root port 3 I/OxApic Enable (PAE) is set
FEC2 8000 - FEC2 FFFF	PCIe port 4	PCIe root port 4 I/OxApic Enable (PAE) is set
FEC4 0000 - FEC4 7FFF	PCIe port 7	PCIe root port 7 I/OxApic Enable (PAE) is set
FEC5 0000 - FEC5 7FFF	PCIe port 9	PCIe root port 9 I/OxApic Enable (PAE) is set
FEC5 8000 - FEC5 FFFF	PCIe port 10	PCIe root port 10 I/OxApic Enable (PAE) is set
FEC6 0000 - FEC6 7FFF	PCIe port 11	PCIe root port 11 I/OxApic Enable (PAE) is set
FEC6 8000 - FEC6 FFFF	PCIe port 12	PCIe root port 12 I/OxApic Enable (PAE) is set
FEF0 0000 - FFFF FFFF	eSPI or SPI	uCode Patch Region Enable UCPR.UPRE is set
FFC0 0000 - FFC7 FFFF FF80 0000 - FF87 FFFF	eSPI or SPI	Bit 8 in BIOS Decode Enable Register is set
FFC8 0000 - FFCF FFFF FF88 0000 - FF8F FFFF	eSPI or SPI	Bit 9 in BIOS Decode Enable Register is set
FFD0 0000 - FFD7 FFFF FF90 0000 - FF97 FFFF	eSPI or SPI	Bit 10 in BIOS Decode Enable Register is set
FFD8 0000 - FFD7 FFFF FF98 0000 - FF9F FFFF	eSPI or SPI	Bit 11 in BIOS Decode Enable Register is set
FFE0 0000 - FFE7 FFFF FFA0 0000 - FFA7 FFFF	eSPI or SPI	Bit 12 in BIOS Decode Enable Register is set
FFE8 0000 - FFEF FFFF FFA8 0000 - FFAF FFFF	eSPI or SPI	Bit 13 in BIOS Decode Enable Register is set
FFF0 0000 - FFF7 FFFF FFB0 0000 - FFB7 FFFF	eSPI or SPI	Bit 14 in BIOS Decode Enable Register is set
FFFC 0000 - FFFF FFFF	eSPI, SPI, or Intel® CSE	Always enabled.

*continued...*



Memory Range	Target	Dependency/Comments
		Refer to <a href="#">Table 15</a> on page 42 for swappable ranges
FFF8 0000 - FFFB FFFF FFB8 0000 - FFBF FFFF	eSPI or SPI	Always enabled. Refer to <a href="#">Table 15</a> on page 42 for swappable ranges
FF70 0000 - FF7F FFFF FF30 0000 - FF3F FFFF	eSPI or SPI	Bit 3 in BIOS Decode Enable Register is set
FF60 0000 - FF6F FFFF FF20 0000 - FF2F FFFF	eSPI or SPI	Bit 2 in BIOS Decode Enable Register is set
FF50 0000 - FF5F FFFF FF10 0000 - FF1F FFFF	eSPI or SPI	Bit 1 in BIOS Decode Enable Register is set
FF40 0000 - FF4F FFFF FF00 0000 - FF0F FFFF	eSPI or SPI	Bit 0 in BIOS Decode Enable Register is set
FED0 X000 - FED0 X3FF	HPET	BIOS determines "fixed" location which is one of four 1 KB ranges where X (in the first column) is 0h, 1h, 2h, or 3h
FED4 0000 - FED4 7FFF	SPI (set by strap)	TPM and Trusted Mobile KBC
FED4 C000 - FED4 FFFF	PCH Internal (PSF Error Handler)	Always enabled
FED6 0000 - FED6 1FFF	PCH Internal (Intel® Trace Hub (Intel® TH)/xHCI)	Always enabled
FED6 2000 - FED6 3FFF	xHCI (CPU )	Fixed range in CPU - never forwarded to PCH
FED5 0000 - FED5 FFFF	Intel® CSE	Always enabled
FED7 0000 - FED7 4FFF	Internal Device	Security feature related
64 KB anywhere in 64-bit address range	USB Host Controller	Enable via standard PCI mechanism (Device 20, Function 0)
2 MB anywhere in 4 GB range	USB Device Controller	Enable via standard PCI mechanism (Device 20, Function 1)
24 KB anywhere in 4 GB range	USB Device Controller	Enable via standard PCI mechanism (Device 20, Function 1)
16 KB anywhere in 64-bit addressing space	Intel® HD Audio Subsystem	Enable via standard PCI mechanism (Device 31, Function 3)
4 KB anywhere in 64-bit addressing space	Intel® HD Audio Subsystem	Enable via standard PCI mechanism (Device 31, Function 3)
64 KB anywhere in 64-bit addressing space	Intel® HD Audio Subsystem	Enable via standard PCI mechanism (Device 31, Function 3)
32 Bytes anywhere in 64-bit address range	SMBus	Enable via standard PCI mechanism (Device 31: Function 4)
2 KB anywhere above 64 KB to 4 GB range	SATA Host Controller	AHCI memory-mapped registers. Enable via standard PCI mechanism (Device 23: Function 0)
Memory Base/Limit anywhere in 4 GB range	PCIe port N (N=1 to 4, 7, 9 to 12)	Enable via standard PCI mechanism
Prefetchable Memory Base/Limit anywhere in 64-bit address range	PCIe port N (N=1 to 4, 7, 9 to 12)	Enable via standard PCI mechanism
16 Bytes anywhere in 64-bit address range	Intel® CSEI #1, #2, #3, #4	Enable via standard PCI mechanism

**continued...**

Memory Range	Target	Dependency/Comments
16 MB anywhere in 64-bit address range	P2SB	Enable via standard PCI mechanism
Eight 4 KB slots anywhere in 64-bit address range	UART, GPI and I2C controllers	Enable via standard PCI mechanism
1 MB (BAR0) or 4 KB (BAR1) in 4GB range	Integrated Sensor Hub	Enable via standard PCI mechanism (Device 19: Function 0)
8 KB slot anywhere in 4 GB range	Integrated Wi-Fi*	Enable via standard PCI mechanism
8 KB slot and 4 KB slot anywhere in 4 GB range	PMC	Enable via standard PCI mechanism
8 KB slot and 4 KB slot anywhere in 4 GB range	Shared SRAM	Enable via standard PCI mechanism

### 4.2.1 Boot Block Update Scheme

The PCH supports a “Top-Block Swap” mode that has the PCH swap the top block in the SPI flash (the boot block) with another location. This allows for safe update of the Boot Block (even if a power failure occurs).

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 15. Boot Block Update Scheme**

BOOT_BLOCK_SIZE Value	Accesses to	Being Directed to
000 (64KB)	FFFF_0000h - FFFF_FFFFh	FFFE_0000h - FFFE_FFFFh and vice versa
001 (128KB)	FFFE_0000h - FFFF_FFFFh	FFFC_0000h - FFFD_FFFFh and vice versa
010 (256KB)	FFFC_0000h - FFFF_FFFFh	FFF8_0000h - FFFB_FFFFh and vice versa
011 (512KB)	FFF8_0000h - FFFF_FFFFh	FFF0_0000h - FFF7_FFFFh and vice versa
100 (1MB)	FFF0_0000h - FFFF_FFFFh	FFE0_0000h - FFEF_FFFFh and vice versa
101 (2MB)	FFE0_0000h - FFFF_FFFFh	FFC0_0000h - FFDF_FFFFh and vice versa
110 (4MB)	FFC0_0000h - FFFF_FFFFh	FF80_0000h - FFBF_FFFFh and vice versa
111 (8MB)	FF80_0000h - FFFF_FFFFh	FF00_0000h - FF7F_FFFFh and vice versa

*Note:* This bit is automatically set to 0 by RTCRST#, but not by PLTRST#.

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

The algorithm is:

1. Software copies the top block to the block immediately below the top
2. Software checks that the copied block is correct. This could be done by performing a checksum calculation.
3. Software sets the “Top-Block Swap” bit. This will invert the appropriate address bits for the cycles going to eSPI or SPI.
4. Software erases the top block

5. Software writes the new top block
6. Software checks the new top block
7. Software clears the top-block swap bit
8. Software sets the Top\_Swap Lock-Down bit

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

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

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

The algorithm to put in the BIOS spec is as follows:

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

## 5.0 Pin Straps

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

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

**Table 16. Pin Straps**

Signal	Usage	When Sampled	Comment
<b>GPP_B14 / SPKR / TIME_SYNC1 / SATA_LED# / ISH_GP6</b>	Top Swap Override	Rising edge of PCH_PWROK	The strap has a 20 kohm ± 30% internal pull-down. 0=>Disable "Top Swap" mode. (Default) 1=>Enable "Top Swap" mode. This inverts an address on access to SPI, so the alternate boot block is fetch instead of the original boot-block. The PCH will invert A16 (default) or the appropriate address lines (A[23:16]) as selected in Top Swap Block size soft strap. <i>Notes:</i> 1. The internal pull-down is disabled after PCH_PWROK is high. 2. Software will not be able to clear the Top Swap bit until the system is rebooted. 3. The status of this strap is readable using the Top Swap bit (Bus0, Device31, Function0, offset DCh, bit4). 4. This signal is in the primary well.
<b>GPP_B18 / ADR_COMPLETE</b>	No Reboot	Rising edge of PCH_PWROK	The strap has a 20 kohm ± 30% internal pull-down. 0=>Disable "No Reboot" mode. (Default) 1=>Enable "No Reboot" mode (PCH will disable the TCO Timer system reboot feature). This function is useful when running ITP/XDP. <i>Notes:</i> 1. The internal pull-down is disabled after PCH_PWROK is high. 2. This signal is in the primary well.
<b>GPP_C2 / SMBALERT#</b>	TLS Confidentiality	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0=>Disable Intel® CSE Crypto Transport Layer Security (TLS) cipher suite (no confidentiality). (Default) 1=>Enable Intel® CSE Crypto Transport Layer Security (TLS) cipher suite (with confidentiality). <i>Notes:</i> 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
<b>GPP_C5 / SMLOALERT#</b>	Boot Strap 0	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. This is bit 0 (LSB) of a total of 4-bit encoded pin straps for boot configuration.

*continued...*

Signal	Usage	When Sampled	Comment
			<p>This strap is used in conjunction with Boot Strap 1,2,3, (on GPP_H0, GPP_H1, GPP_H2 respectively).</p> <p>4-bit boot strap configuration encodings:</p> <p>0000 = Initiator Attached Flash Configuration (BIOS / CSE on SPI). eSPI is enabled</p> <p>0010 = Initiator Attached Flash Configuration (BIOS / CSE on SPI). eSPI is disabled</p> <p>0100 = BIOS on eSPI Peripheral Channel; CSE on initiator attached SPI</p> <p>1000 = Target Attached Flash Configuration (BIOS / CSE on eSPI attached device).</p> <p>1100 = BIOS on eSPI peripheral Channel; CSE on target attached SPI.</p> <p>Others: Reserved</p> <p>Notes: 1. The internal pull-down is disabled after RSMRST# de-asserts.</p> <p>2. This signal is in the primary well.</p>
<b>SPIO_MOSI</b>	Reserved	Rising edge of RSMRST#	<p>External pull-up is required. Recommend 4.7 kohm pull up.</p> <p>This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.</p>
<b>GPP_D10 / ISH_SPI_CLK / BSSB_LS2_TX / GSPI2_CLK</b>	BSSB_LS2 pins VCC configuration	Rising edge of RSMRST#	<p>This strap has a 20 kohm <math>\pm</math> 30% internal pull-down.</p> <p>0 = BSSB_LS2 pins at 1.8 V</p> <p>1 = BSSB_LS2 pins at 3.3 V</p> <p>Notes: 1. The internal pull-down is disabled after RSMRST# de-asserts.</p> <p>2. This signal is in the primary well.</p>
<b>GPP_D12 / ISH_SPI_MOSI / BSSB_LS3_TX / GSPI2_MOSI</b>	BSSB_LS3 pins VCC configuration	Rising edge of RSMRST#	<p>This strap has a 20 kohm <math>\pm</math> 30% internal pull-down.</p> <p>0 = BSSB_LS3 pins at 1.8 V</p> <p>1 = BSSB_LS3 pins at 3.3 V</p> <p>Notes: 1. The internal pull-down is disabled after RSMRST# de-asserts.</p> <p>2. This signal is in the primary well.</p>
<b>GPP_B23 / SMLIALERT# / PCHHOT#</b>	CPUNSSC Clock Frequency	Rising edge of RSMRST#	<p>This strap has a 20 kohm <math>\pm</math> 30% internal pull-down.</p> <p>0 = 38.4 MHz clock (direct from crystal) (default)</p> <p>1 = 19.2 MHz clock (derived from 38.4 MHz crystal)</p> <p>Notes: 1. The internal pull-down is disabled after RSMRST# de-asserts.</p> <p>2. When used as PCHHOT# and strap low, a 150 kohm pull-up is needed to ensure it does not override the internal pull-down strap sampling.</p> <p>3. This signal is in the primary well.</p>
<b>SPIO_IO2</b>	Reserved	Rising edge of RSMRST#	<p>External pull-up is required. Recommend 100 kohm if pulled up to 3.3 V or 75 kohm if pulled up to 1.8 V.</p> <p>This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.</p>
<b>SPIO_IO3</b>	Reserved	Rising edge of RSMRST#	<p>External pull-up is required. Recommend 100 kohm if pulled up to 3.3 V or 75 kohm if pulled up to 1.8 V.</p> <p>This strap should sample HIGH. There should NOT be any on-board device driving it to opposite direction during strap sampling.</p>

*continued...*


Signal	Usage	When Sampled	Comment
<b>GPP_R2 / HDA_SDO / I2S0_TXD / HDACPU_SDO</b>	Flash Descriptor Security Override	Rising edge of PCH_PWROK	This strap has a 20 kohm ± 30% internal pull-down. 0=> Enable security measures defined in the Flash Descriptor. (Default) 1=> Disable Flash Descriptor Security ( <i>override</i> ). This strap should only be asserted high using external Pull-up in manufacturing/debug environments ONLY. <i>Notes:</i> 1. The internal pull-down is disabled after PCH_PWROK is high. 2. This signal is in the primary well.
<b>GPP_E6 / THC0_SPI1_RST#</b>	JTAG ODT Disable	Rising edge of RSMRST#	This strap does not have an internal pull-up or pull-down. External pull-up is recommended 0=> JTAG ODT is disabled 1=> JTAG ODT is enabled
<b>GPP_E19 / DDP1_CTRLDATA / BSSB_LS0_TX</b>	DDP1 I2C / BSSB_LS0 pins VCC configuration	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0=> DDP1 I2C / BSSB_LS0 pins at 1.8 V 1=> DDP1 I2C / BSSB_LS0 pins at 3.3 V <i>Notes:</i> 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
<b>GPP_E21 / DDP2_CTRLDATA / BSSB_LS1_RX</b>	DDP2 I2C / BSSB_LS1 pins VCC configuration	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0 = DDP2 I2C / BSSB LS1 pins at 1.8 V 1 = DDP2 I2C / BSSB LS1 pins at 3.3 V <i>Notes:</i> 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
<b>DBG_PMODE</b>	Reserved	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-up. This strap should sample high. There should NOT be any on- board device driving it to opposite direction during strap sampling. <i>Notes:</i> 1. The internal pull-up is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
<b>GPD7</b>	Reserved	Rising edge of DSW_PWROK	This strap has a 20 kohm ± 30% internal pull-down. This strap should sample LOW. There should NOT be any on- board device driving it to opposite direction during strap sampling. <i>Notes:</i> 1. The internal pull-down is disabled after DSW_PWROK is high. 2. This signal is in the DSW well.
<b>GPP_F0 / CNV_BRI_DT / UART2_RTS#</b>	XTAL Frequency Selection	Rising edge of RSMRST#	This strap has a 20 kohm ± 30% internal pull-down. 0 = 38.4 MHz (default) 1 = 24 MHz <i>Notes:</i> 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
<b>GPP_F2 / CNV_RGI_DT / UART2_TXD</b>	M.2 CNVi Mode Select	Rising edge of RSMRST#	This strap does not have an internal pull-up or pull-down. A weak external pull-up is required. 0=>Integrated CNVi enabled. 1=>Integrated CNVi disabled. <i>Note:</i> When a RF companion chip is connected to the PCH CNVi interface, the device internal pull-down resistor will pull the strap low to enable CNVi interface.

*continued...*

Signal	Usage	When Sampled	Comment
<b>GPP_F7</b>	Reserved	Rising edge of RSMRST#	This strap has a 20 kohm $\pm$ 30% internal pull-down. This strap should sample LOW. There should NOT be any on-board device driving it to opposite direction during strap sampling. <i>Notes:</i> 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
<b>GPP_F10</b>	Reserved	Rising edge of RSMRST#	This strap has a 20 kohm $\pm$ 30% internal pull-down. This strap should sample LOW. There should NOT be any on-board device driving it to opposite direction during strap sampling. <i>Notes:</i> 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
<b>GPP_H0</b>	Boot Strap 1	Rising edge of RSMRST#	This strap has a 20 kohm $\pm$ 30% internal pull-down. This is bit 1 of a total of 4-bit encoded pin straps for boot configuration. Refer to Boot Strap 0 (on GPP_C5) for the encoding. <i>Notes:</i> 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
<b>GPP_H1</b>	Boot Strap 2	Rising edge of RSMRST#	This strap has a 20 kohm $\pm$ 30% internal pull-down. This is bit 2 of a total of 4-bit encoded pin straps for boot configuration. Refer to Boot Strap 0 (on GPP_C5) for the encoding. <i>Notes:</i> 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
<b>GPP_H2</b>	Boot Strap 3	Rising edge of RSMRST#	This strap has a 20 kohm $\pm$ 30% internal pull-down. This is bit 3 of a total of 4-bit encoded pin straps for boot configuration. Refer to Boot Strap 0 (on GPP_C5) for the encoding. <i>Notes:</i> 1. The internal pull-down is disabled after RSMRST# de-asserts. 2. This signal is in the primary well.
<b>SPIVCCIOSEL</b>	SPI Operation Voltage Select	Not Sampled. This strap must always be driven to a valid logic level	There is no internal pull-up or pull-down on the strap. An external resistor is required. 0 = SPI voltage is 3.3 V (4.7 kohm pull-down to GND) 1 = SPI voltage is 1.8 V (4.7 kohm pull-up to VCCDSW_3P3)

## 6.0 Electrical and Thermal Characteristics

---

For information on the Electrical and Thermal Characteristics, refer to download the pdf, click  on the navigation pane and refer the spreadsheet, **759603\_001\_Electr\_Therm\_Spec.xlsx**



## 7.0 Technologies

---

This chapter provides a high-level description of Intel technologies implemented in the processor.

The implementation of the features may vary between the processor SKUs.

Details on the different technologies of Intel processors and other relevant external notes are located at the Intel technology web site: <http://www.intel.com/technology/>

### 7.1 Platform Environmental Control Interface (PECI)

PECI is an Intel proprietary interface that provides a communication channel between Intel processors and external components such as Super IO (SIO) and Embedded Controllers (EC) to provide processor temperature, Turbo, Configurable TDP, and Memory Throttling Control mechanisms and many other services. PEFI is used for platform thermal management and real-time control and configuration of processor features and performance.

---

**NOTE**

PECI over eSPI is supported.

---

#### 7.1.1 PEFI Bus Architecture

The PEFI architecture is based on a wired-OR bus that the clients (as processor PEFI) can pull up (with the strong drive).

The idle state on the bus is '0' (logical low) and near zero (Logical voltage level).

---

**NOTE**

PECI supported frequency range is 3.2 kHz - 1 MHz.

---

The following figures demonstrate PEFI design and connectivity:

- PEFI Host-Clients Connection: While the host/originator can be third party PEFI host and one of the PEFI client is a processor PEFI device.
- PEFI EC Connection.

Figure 3. Example for PECE Host-Clients Connection

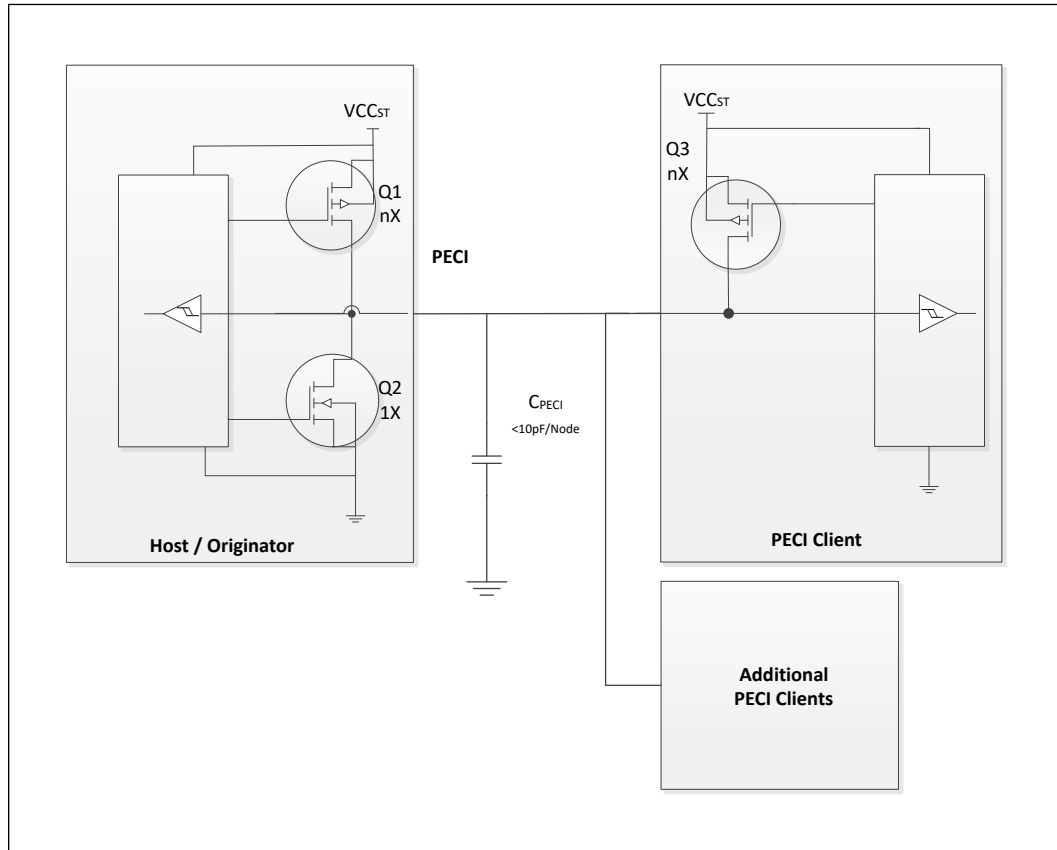
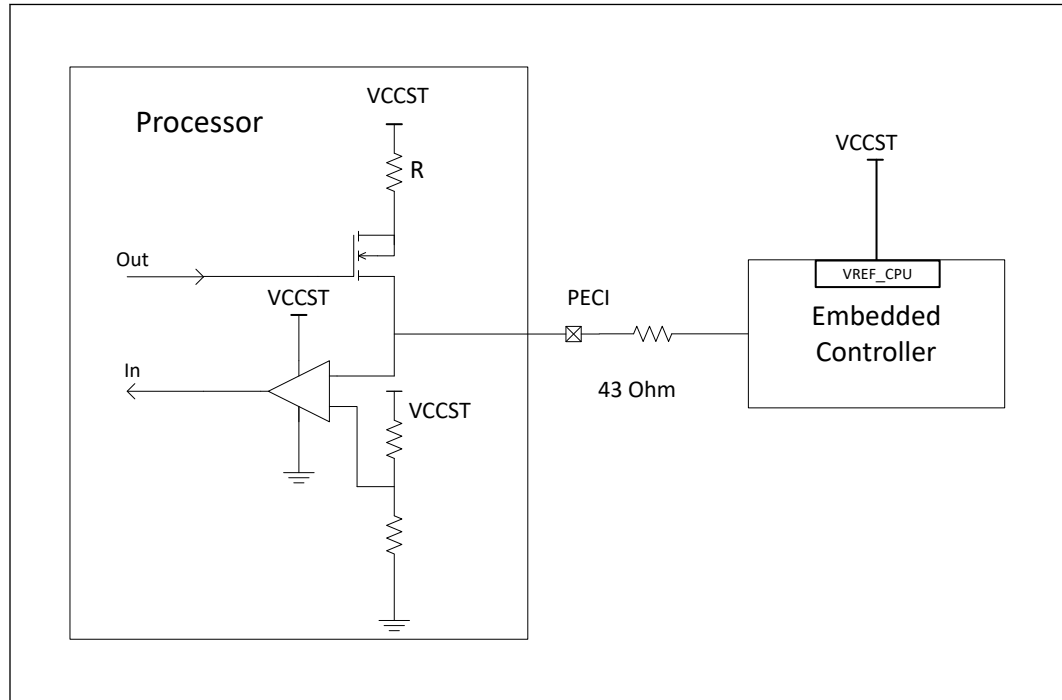


Figure 4. Example for PECI EC Connection



## 7.2 Intel® Virtualization Technology (Intel® VT)

Intel® Virtualization Technology (Intel® VT) makes a single system appear as multiple independent systems to software. This allows multiple, independent operating systems to run simultaneously on a single system. Intel® VT comprises technology components to support Virtualization of platforms based on Intel® architecture microprocessors and chipsets.

Intel® Virtualization Technology (Intel® VT) Intel® 64 and Intel® Architecture (Intel® VT-x) added hardware support in the processor to improve the Virtualization performance and robustness. Intel® Virtualization Technology for Directed I/O (Intel® VT-d) extends Intel® VT-x by adding hardware assisted support to improve I/O device Virtualization performance.

Intel® VT-x specifications and functional descriptions are included in the *Intel® 64 Architectures Software Developer's Manual, Volume 3*. Available at:

<http://www.intel.com/products/processor/manuals>

The Intel® VT-d specification and other VT documents can be referenced at:

<http://www.intel.com/content/www/us/en/virtualization/virtualization-technology/>.

## 7.2.1 Intel® Virtualization Technology (Intel® VT) for Intel® 64 and Intel® Architecture (Intel® VT-x)

### Intel® VT-x Objectives

Intel® VT-x provides hardware acceleration for virtualization of IA platforms. Virtual Machine Monitor (VMM) can use Intel® VT-x features to provide an improved reliable Virtualization platform. By using Intel® VT-x, a VMM is:

- **Robust:** VMMs no longer need to use para-virtualization or binary translation. This means that VMMs will be able to run off-the-shelf operating systems and applications without any special steps.
- **Enhanced:** Intel® VT enables VMMs to run 64-bit guest operating systems on IA x86 processors.
- **More Reliable:** Due to the hardware support, VMMs can now be smaller, less complex, and more efficient. This improves reliability and availability and reduces the potential for software conflicts.
- **More Secure:** The use of hardware transitions in the VMM strengthens the isolation of VMs and further prevents corruption of one VM from affecting others on the same system.

### Intel® VT-x Key Features

The processor supports the following added new Intel® VT-x features:

- **Mode-based Execute Control for EPT (MBEC)** - A mode of EPT operation which enables different controls for executability of Guest Physical Address (GPA) based on Guest specified mode (User/ Supervisor) of linear address translating to the GPA. When the mode is enabled, the executability of a GPA is defined by two bits in EPT entry. One bit for accesses to user pages and other one for accesses to supervisor pages.
  - This mode requires changes in VMCS and EPT entries. VMCS includes a bit "Mode-based execute control for EPT" which is used to enable/disable the mode. An additional bit in EPT entry is defined as "execute access for user-mode linear addresses"; the original EPT execute access bit is considered as "execute access for supervisor-mode linear addresses". If the "mode-based execute control for EPT" VM-execution control is disabled the additional bit is ignored and the system work with one bit i.e. the original bit, for execute control for both user and supervisor pages.
  - Behavioral changes - Behavioral changes are across three areas:
    - **Access to GPA** - If the "Mode-based execute control for EPT" VMexecution control is 1, treatment of guest-physical accesses by instruction fetches depends on the linear address from which an instruction is being fetched.
      1. If the translation of the linear address specifies user mode (the U/S bit was set in every paging structure entry used to translate the linear address), the resulting guest-physical address is executable under EPT only if the XU bit (at position 10) is set in every EPT paging-structure entry used to translate the guest-physical address.

2. If the translation of the linear address specifies supervisor mode (the U/ S bit was clear in at least one of the paging-structure entries used to translate the linear address), the resulting guest-physical address is executable under EPT only if the XS bit is set in every EPT paging-structure entry used to translate the guest-physical address.
  - The XU and XS bits are used only when translating linear addresses for guest code fetches. They do not apply to guest page walks, data accesses, or A/D-bit updates.
- **VMEntry** - If the "activate secondary controls" and "Mode-based execute control for EPT" VM-execution controls are both 1, VM entries ensure that the "enable EPT" VM-execution control is 1. VM entry fails if this check fails. When such a failure occurs, control is passed to the next instruction.
- **VMExit** - The exit qualification due to EPT violation reports clearly whether the violation was due to User mode access or supervisor mode access.
  - Capability Querying: IA32\_VMX\_PROCBASED\_CTL2 has bit to indicate the capability, RDMSR can be used to read and query whether the processor supports the capability or not.
- Extended Page Table (EPT) Accessed and Dirty Bits
  - EPT A/D bits enabled VMMs to efficiently implement memory management and page classification algorithms to optimize VM memory operations, such as defragmentation, paging, live migration, and check-pointing. Without hardware support for EPT A/D bits, VMMs may need to emulate A/D bits by marking EPT paging-structures as not-present or read-only, and incur the overhead of EPT page-fault VM exits and associated software processing.
- EPTP (EPT pointer) switching
  - EPTP switching is a specific VM function. EPTP switching allows guest software (in VMX non-root operation, supported by EPT) to request a different EPT paging-structure hierarchy. This is a feature by which software in VMX non-root operation can request a change of EPTP without a VM exit. The software will be able to choose among a set of potential EPTP values determined in advance by software in VMX root operation.
- Pause loop exiting
  - Support VMM schedulers seeking to determine when a virtual processor of a multiprocessor virtual machine is not performing useful work. This situation may occur when not all virtual processors of the virtual machine are currently scheduled and when the virtual processor in question is in a loop involving the PAUSE instruction. The new feature allows detection of such loops and is thus called PAUSE-loop exiting.

The processor IA core supports the following Intel® VT-x features:

- Extended Page Tables (EPT)
  - EPT is hardware assisted page table virtualization
  - It eliminates VM exits from guest OS to the VMM for shadow page-table maintenance
- Virtual Processor IDs (VPID)
  - Ability to assign a VM ID to tag processor IA core hardware structures (such as TLBs)

- This avoids flushes on VM transitions to give a lower-cost VM transition time and an overall reduction in virtualization overhead.
- Guest Preemption Timer
  - The mechanism for a VMM to preempt the execution of a guest OS after an amount of time specified by the VMM. The VMM sets a timer value before entering a guest
  - The feature aids VMM developers in flexibility and Quality of Service (QoS) guarantees
- Descriptor-Table Exiting
  - Descriptor-table exiting allows a VMM to protect a guest OS from internal (malicious software based) attack by preventing the relocation of key system data structures like IDT (interrupt descriptor table), GDT (global descriptor table), LDT (local descriptor table), and TSS (task segment selector).
  - A VMM using this feature can intercept (by a VM exit) attempts to relocate these data structures and prevent them from being tampered by malicious software.

## 7.2.2 Intel® VT for Directed I/O

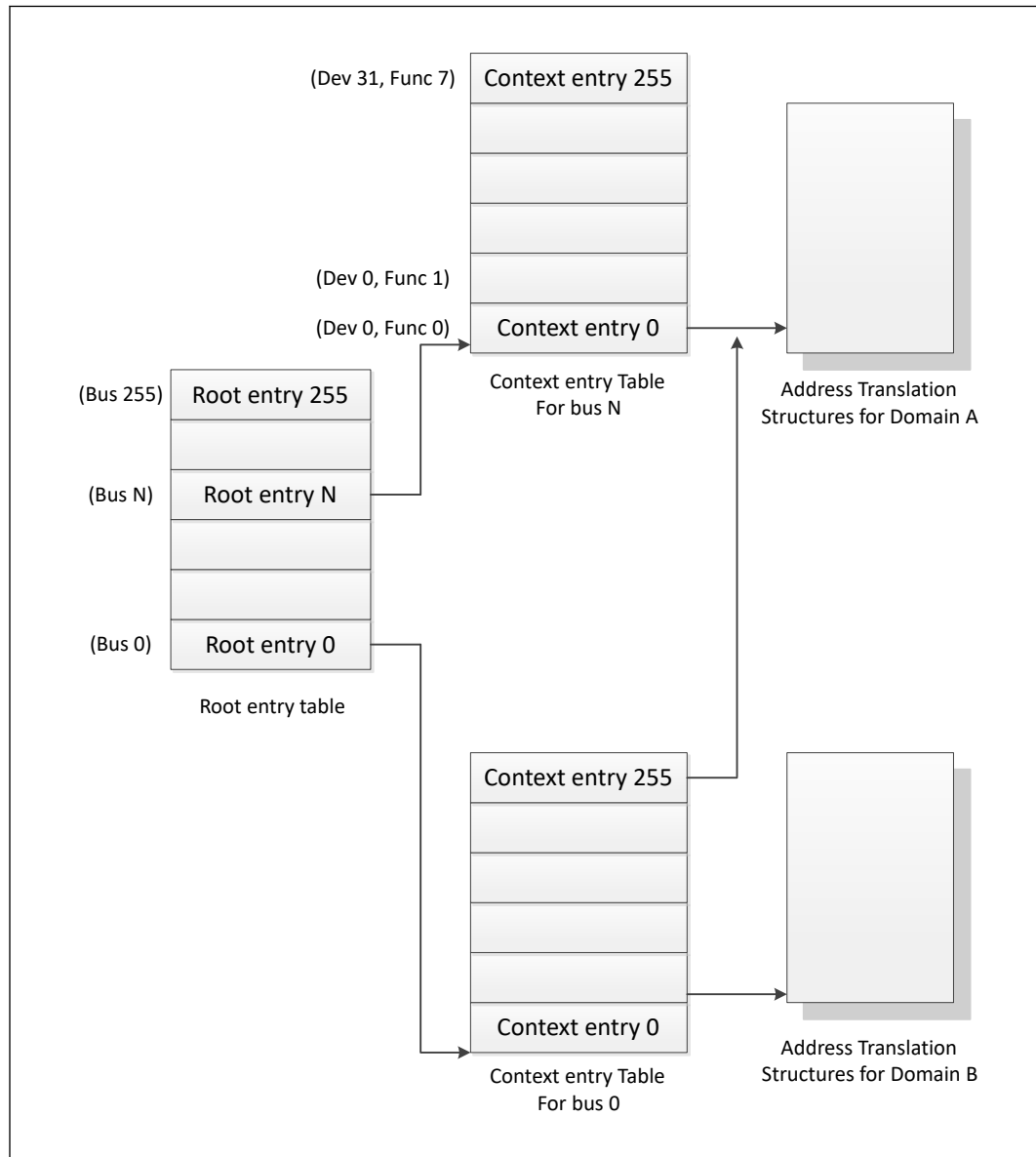
### Intel® VT-d Objectives

The Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d) objectives are domain-based isolation and hardware-based virtualization. A domain can be abstractly defined as an isolated environment in a platform to which a subset of host physical memory is allocated. Intel® VT-d provides accelerated I/O performance for a Virtualization platform and provides software with the following capabilities:

- **I/O Device Assignment and Security:** for flexibly assigning I/O devices to VMs and extending the protection and isolation properties of VMs for I/O operations.
- **DMA Remapping:** for supporting independent address translations for Direct Memory Accesses (DMA) from devices.
- **Interrupt Remapping:** for supporting isolation and routing of interrupts from devices and external interrupt controllers to appropriate VMs.
- **Reliability:** for recording and reporting to system software DMA and interrupt errors that may otherwise corrupt memory or impact VM isolation.

Intel® VT-d accomplishes address translation by associating transaction from a given I/O device to a translation table associated with the Guest to which the device is assigned. It does this by means of the data structure in the following illustration. This table creates an association between the device's PCI Express\* Bus/Device/Function (B/D/F) number and the base address of a translation table. This data structure is populated by a VMM to map devices to translation tables in accordance with the device assignment restrictions above and to include a multi-level translation table (VT-d Table) that contains Guest specific address translations.

Figure 5. Device to Domain Mapping Structures



Intel® VT-d functionality often referred to as an Intel® VT-d Engine, has typically been implemented at or near a PCI Express\* host bridge component of a computer system. This might be in a chipset component or in the PCI Express functionality of a processor with integrated I/O. When one such VT-d engine receives a PCI Express transaction from a PCI Express bus, it uses the B/D/F number associated with the transaction to search for an Intel® VT-d translation table. In doing so, it uses the B/D/F number to traverse the data structure shown in the above figure. If it finds a valid Intel® VT-d table in this data structure, it uses that table to translate the address provided on the PCI Express bus. If it does not find a valid translation table for a given translation, this results in an Intel® VT-d fault. If Intel® VT-d translation is required, the Intel® VT-d engine performs an N-level table walk.

For more information, refer to *Intel® Virtualization Technology for Directed I/O Architecture Specification* <http://www.intel.com/content/dam/www/public/us/en/documents/product-specifications/vt-directed-io-spec.pdf>

### Intel® VT-d Key Features

The processor supports the following Intel® VT-d features:

- Memory controller and processor graphics comply with the Intel® VT-d 2.1 Specification.
- Two Intel® VT-d DMA remap engines.
  - iGFX DMA remap engine
  - Default DMA remap engine (covers all devices except iGFX)
- Support for root entry, context entry, and the default context
- 46-bit guest physical address and host physical address widths
- Support for 4K page sizes only
- Support for register-based fault recording only (for single entry only) and support for MSI interrupts for faults
- Support for both leaf and non-leaf caching
- Support for boot protection of default page table
- Support for non-caching of invalid page table entries
- Support for hardware-based flushing of translated but pending writes and pending reads, on IOTLB invalidation
- Support for Global, Domain-specific and Page specific IOTLB invalidation
- MSI cycles (MemWr to address FEEx\_xxxxh) not translated.
- Interrupt Remapping is supported
- Queued invalidation is supported
- Intel® VT-d translation bypass address range is supported (Pass Through)

The processor supports the following added new Intel® VT-d features:

- 4-level Intel® VT-d Page walk – both default Intel® VT-d engine, as well as the Processor Graphics VT-d engine are upgraded to support 4-level Intel® VT-d tables (adjusted guest address width of 48 bits)
- Intel® VT-d super-page – support of Intel® VT-d super-page (2 MB, 1 GB) for default Intel® VT-d engine (that covers all devices except IGD)  
IGD Intel® VT-d engine does not support super-page and BIOS should disable super-page in default Intel® VT-d engine when iGfx is enabled.

### 7.2.3 Intel® APIC Virtualization Technology

Intel® APIC Virtualization Technology (Intel® APICv) is a collection of features that can be used to support the virtualization of interrupts and the Advanced Programmable Interrupt Controller (APIC).

When APIC virtualization is enabled, the processor emulates many accesses to the APIC, tracks the state of the virtual APIC, and delivers virtual interrupts — all in VMX non-root operation without a VM exit.



The following are the VM-execution controls relevant to APIC virtualization and virtual interrupts:

- **Virtual-interrupt Delivery.** This control enables the evaluation and delivery of pending virtual interrupts. It also enables the emulation of writes (memory-mapped or MSR-based, as enabled) to the APIC registers that control interrupt prioritization.
- **Use TPR Shadow.** This control enables emulation of accesses to the APIC's task-priority register (TPR) via CR8 and, if enabled, via the memory-mapped or MSR-based interfaces.
- **Virtualize APIC Accesses.** This control enables virtualization of memory-mapped accesses to the APIC by causing VM exits on accesses to a VMM-specified APIC-access page. Some of the other controls, if set, may cause some of these accesses to be emulated rather than causing VM exits.
- **Virtualize x2APIC Mode.** This control enables virtualization of MSR-based accesses to the APIC.
- **APIC-register Virtualization.** This control allows memory-mapped and MSR-based reads of most APIC registers (as enabled) by satisfying them from the virtual-APIC page. It directs memory-mapped writes to the APIC-access page to the virtual-APIC page, following them by VM exits for VMM emulation.
- **Process Posted Interrupts.** This control allows software to post virtual interrupts in a data structure and send a notification to another logical processor; upon receipt of the notification, the target processor will process the posted interrupts by copying them into the virtual-APIC page.

Intel® APIC Virtualization specifications and functional descriptions are included in the *Intel® 64 Architectures Software Developer's Manual, Volume 3*, available at:

<http://www.intel.com/products/processor/manuals>

## 7.3 Security Technologies

### 7.3.1 Intel® Advanced Encryption Standard New Instructions

The processor supports Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI) that are a set of Single Instruction Multiple Data (SIMD) instructions that enable fast and secure data encryption and decryption based on the Advanced Encryption Standard (AES). Intel® AES-NI is valuable for a wide range of cryptographic applications, such as applications that perform bulk encryption/decryption, authentication, random number generation, and authenticated encryption. AES is broadly accepted as the standard for both government and industrial applications and is widely deployed in various protocols.

Intel® AES-NI consists of six Intel® SSE instructions. Four instructions, AESENC, AESENCLAST, AESDEC, and AESDELAST facilitate high-performance AES encryption and decryption. The other two, AESIMC and AESKEYGENASSIST, support the AES key expansion procedure. Together, these instructions provide full hardware for supporting AES; offering security, high performance, and a great deal of flexibility.

This generation of the processor has increased the performance of the Intel® AES-NI significantly compared to previous products.

The Intel® AES-NI specifications and functional descriptions are included in the *Intel® 64 Architectures Software Developer's Manual, Volume 2*, available at:

<http://www.intel.com/products/processor/manuals>

### 7.3.2 Perform Carry-Less Multiplication Quad Word Instruction (PCLMULQDQ)

The processor supports the carry-less multiplication instruction, PCLMULQDQ. PCLMULQDQ is a Single Instruction Multiple Data (SIMD) instruction that computes the 128-bit carry-less multiplication of two 64-bit operands without generating and propagating carries. Carry-less multiplication is an essential processing component of several cryptographic systems and standards. Hence, accelerating carry-less multiplication can significantly contribute to achieving high-speed secure computing and communication.

PCLMULQDQ specifications and functional descriptions are included in the *Intel® 64 Architectures Software Developer's Manual, Volume 2*. Available at:

<http://www.intel.com/products/processor/manuals>

### 7.3.3 Intel® Secure Key

The processor supports Intel® Secure Key (formerly known as Digital Random Number Generator or DRNG), a software visible random number generation mechanism supported by a high-quality entropy source. This capability is available to programmers through the RDRAND instruction. The resultant random number generation capability is designed to comply with existing industry standards in this regard (ANSI X9.82 and NIST SP 800-90).

Some possible usages of the RDRAND instruction include cryptographic key generation as used in a variety of applications, including communication, digital signatures, secure storage, etc.

RDRAND specifications and functional descriptions are included in the *Intel® 64 Architectures Software Developer's Manual, Volume 2*. Available at:

<http://www.intel.com/products/processor/manuals>

### 7.3.4 Execute Disable Bit

The Execute Disable Bit allows memory to be marked as non-executable when combined with a supporting operating system. If code attempts to run in non-executable memory, the processor raises an error to the operating system. This feature can prevent some classes of viruses or worms that use buffer overrun vulnerabilities and can, thus, help improve the overall security of the system.

### 7.3.5 Boot Guard Technology

Boot Guard technology is a part of boot integrity protection technology. Boot Guard can help protect the platform boot integrity by preventing the execution of unauthorized boot blocks. With Boot Guard, platform manufacturers can create boot policies such that invocation of an unauthorized (or untrusted) boot block will trigger the platform protection per the manufacturer's defined policy.

With verification based in the hardware, Boot Guard extends the trust boundary of the platform boot process down to the hardware level.

Boot Guard accomplishes this by:

- Providing of hardware-based Static Root of Trust for Measurement (S-RTM) and the Root of Trust for Verification (RTV) using Intel architectural components.
- Providing of architectural definition for platform manufacturer Boot Policy.
- Enforcing manufacturer provided Boot Policy using Intel architectural components.

Benefits of this protection are that Boot Guard can help maintain platform integrity by preventing re-purposing of the manufacturer's hardware to run an unauthorized software stack.

---

**NOTE**

Boot Guard availability may vary between the different SKUs.

---

### 7.3.6 Intel® Supervisor Mode Execution Protection (SMEP)

Intel® Supervisor Mode Execution Protection (SMEP) is a mechanism that provides the next level of system protection by blocking malicious software attacks from user mode code when the system is running in the highest privilege level. This technology helps to protect from virus attacks and unwanted code from harming the system. For more information, refer to *Intel® 64 Architectures Software Developer's Manual, Volume 3* at:

<http://www.intel.com/products/processor/manuals>

### 7.3.7 Intel® Supervisor Mode Access Protection (SMAP)

Intel® Supervisor Mode Access Protection (SMAP) is a mechanism that provides next level of system protection by blocking a malicious user from tricking the operating system into branching off user data. This technology shuts down very popular attack vectors against operating systems.

For more information, refer to the *Intel® 64 Architectures Software Developer's Manual, Volume 3*:

<http://www.intel.com/products/processor/manuals>

### 7.3.8 Intel® Secure Hash Algorithm Extensions (Intel® SHA Extensions)

The Secure Hash Algorithm (SHA) is one of the most commonly employed cryptographic algorithms. Primary usages of SHA include data integrity, message authentication, digital signatures, and data de-duplication. As the pervasive use of security solutions continues to grow, SHA can be seen in more applications now than ever. The Intel® SHA Extensions are designed to improve the performance of these compute-intensive algorithms on Intel® architecture-based processors.

The Intel® SHA Extensions are a family of seven instructions based on the Intel® Streaming SIMD Extensions (Intel® SSE) that are used together to accelerate the performance of processing SHA-1 and SHA-256 on Intel architecture-based processors. Given the growing importance of SHA in our everyday computing devices, the new instructions are designed to provide a needed boost of performance to hashing a single buffer of data. The performance benefits will not only help improve responsiveness and lower power consumption for a given application, but they may

also enable developers to adopt SHA in new applications to protect data while delivering to their user experience goals. The instructions are defined in a way that simplifies their mapping into the algorithm processing flow of most software libraries, thus enabling easier development.

More information on Intel® SHA can be found at:

<http://software.intel.com/en-us/artTGLes/intel-sha-extensions>

### 7.3.9 User Mode Instruction Prevention (UMIP)

User Mode Instruction Prevention (UMIP) provides additional hardening capability to the OS kernel by allowing certain instructions to execute only in supervisor mode (Ring 0).

If the OS opt-in to use UMIP, the following instruction are enforced to run in supervisor mode:

- **SGDT** - Store the GDTR register value
- **SIDT** - Store the IDTR register value
- **SLDT** - Store the LDTR register value
- **SMSW** - Store Machine Status Word
- **STR** - Store the TR register value

An attempt at such execution in user mode causes a general protection exception (#GP).

UMIP specifications and functional descriptions are included in the *Intel® 64 Architectures Software Developer's Manual, Volume 3*. Available at:

<http://www.intel.com/products/processor/manuals>

### 7.3.10 Read Processor ID (RDPID)

A companion instruction that returns the current logical processor's ID and provides a faster alternative to using the RDTSCP instruction.

RDPID specifications and functional descriptions are included in the *Intel® 64 Architectures Software Developer's Manual, Volume 2*. Available at:

<http://www.intel.com/products/processor/manuals>

### 7.3.11 Control-flow Enforcement Technology (Intel® CET)

Return-oriented Programming (ROP), and similarly CALL/JMP-oriented programming (COP/JOP), have been the prevalent attack methodology for stealth writers targeting vulnerabilities in programs.

CET provides the following components to defend against ROP/JOP style control-flow subversion attacks.

### 7.3.11.1 Shadow Stack

A shadow stack is a second stack for the program that is used exclusively for control transfer operations. This stack is separate from the data stack and can be enabled for operation individually in user mode or supervisor mode.

The shadow stack is protected from tamper through the page table protections such that regular store instructions cannot modify the contents of the shadow stack. To provide this protection the page table protections are extended to support an additional attribute for pages to mark them as “Shadow Stack” pages. When shadow stacks are enabled, control transfer instructions/flows such as near call, far call, call to interrupt/exception handlers, etc. store their return addresses to the shadow stack. The RET instruction pops the return address from both stacks and compares them. If the return addresses from the two stacks do not match, the processor signals a control protection exception (#CP). Stores from instructions such as MOV, XSAVE, etc. are not allowed to the shadow stack.

### 7.3.11.2 Indirect Branch Tracking

The ENDBR32 and ENDBR64 (collectively ENDBRANCH) are two new instructions that are used to mark valid indirect CALL/JMP target locations in the program. This instruction is a NOP on legacy processors for backward compatibility.

The processor implements a state machine that tracks indirect JMP and CALL instructions. When one of these instructions is seen, the state machine moves from IDLE to WAIT\_FOR\_ENDBRANCH state. In WAIT\_FOR\_ENDBRANCH state the next instruction in the program stream must be an ENDBRANCH. If an ENDBRANCH is not seen the processor causes a control protection fault (#CP), otherwise the state machine moves back to IDLE state.

More information on Intel® CET can be found at:

<https://software.intel.com/sites/default/files/managed/4d/2a/control-flow-enforcement-technology-preview.pdf>

### 7.3.12 KeyLocker Technology

A method to make long-term keys short-lived without exposing them. This protects against vulnerabilities when keys can be used to attack encrypted data such as disk drives.

An instruction (LOADIWKEY) allows the OS to load a random wrapping value (IWKey). The IWKey can be backed up and restored by the OS to/from the PCH in a secure manner.

The Software can wrap its own key via the ENCODEKEY instruction and receive a handle. The handle is used with the AES\*KL instructions to handle encrypt and decrypt operations. Once a handle is obtained, the software can delete the original key from memory.

## 7.4 Power and Performance Technologies

### 7.4.1 Intel® Smart Cache Technology

The Intel® Smart Cache Technology is a shared Last Level Cache (LLC).

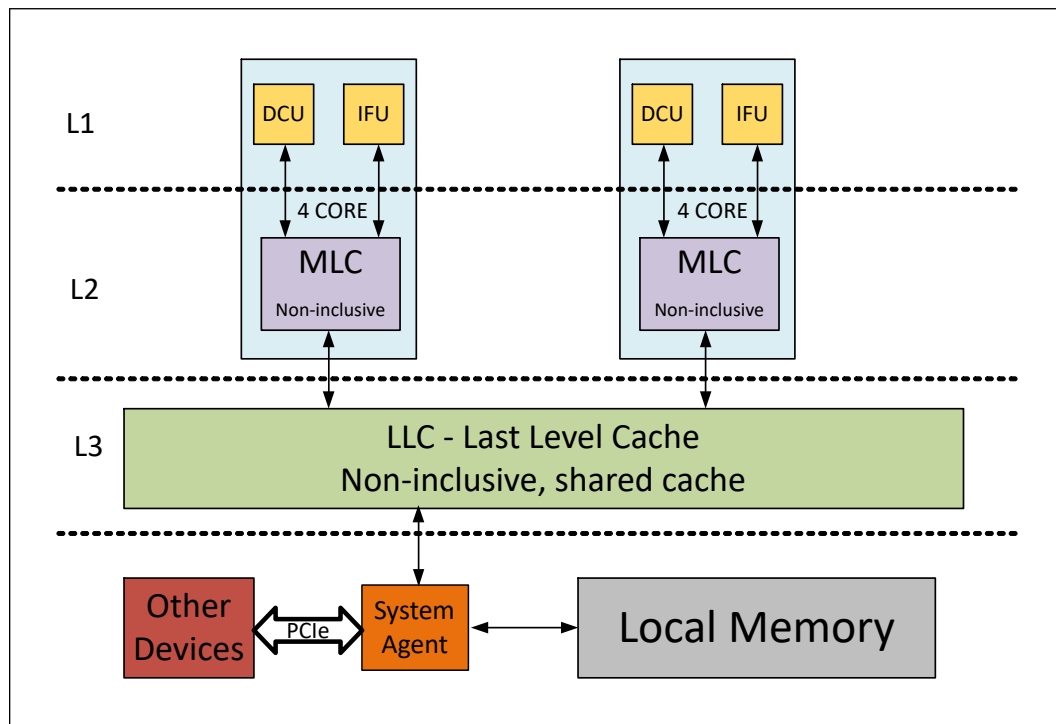
- The LLC is non-inclusive.
- The LLC may also be referred to as a 3rd level cache.
- The LLC is shared between all IA cores as well as the Processor Graphics.
- The 1st and 2nd level caches are not shared between physical cores and each physical core has a separate set of caches.
- The size of the LLC is SKU specific with a maximum of 6MB and is a 12-way associative cache.

### 7.4.2 IA Core Level 1 and Level 2 Caches

The 1st level cache is divided into a data cache (DFU) and an instruction cache (IFU). The processor 1st level cache size is 32KB for data and 64KB for instructions. The 1st level cache is an 8-way associative cache.

The 2nd level cache holds both data and instructions. It is also referred to as mid-level cache or MLC. The processor 2nd level cache size is up to 4 MB and is a 20-way non-inclusive associative cache.

**Figure 6. Processor Cache Hierarchy**



**NOTES**

1. L1 Data cache (DCU) - 32KB (per core)
2. L1 Instruction cache (IFU) - 64KB (per core)
3. MLC - Mid Level Cache - 2MB (per 4 core)
4. LLC - Last level cache - 6MB

### 7.4.3 Ring Interconnect

The Ring is a high speed, wide interconnect that links the processor cores, processor graphics and the System Agent.

The Ring shares frequency and voltage with the Last Level Cache (LLC).

The Ring's frequency dynamically changes. Its frequency is relative to both processor cores and processor graphics frequencies.

### 7.4.4 Power Aware Interrupt Routing (PAIR)

The processor includes enhanced power-performance technology that routes interrupts to threads or processor IA cores based on their sleep states. As an example, for energy savings, it routes the interrupt to the active processor IA cores without waking the deep idle processor IA cores. For performance, it routes the interrupt to the idle (C1) processor IA cores without interrupting the already heavily loaded processor IA cores. This enhancement is most beneficial for high-interrupt scenarios like WLAN peripherals, etc.

### 7.4.5 Enhanced Intel SpeedStep® Technology

Enhanced Intel SpeedStep® Technology enables OS to control and select P-state. The following are the key features of Enhanced Intel SpeedStep® Technology:

- Multiple frequencies and voltage points for optimal performance and power efficiency. These operating points are known as P-states.
- Frequency selection is software controlled by writing to processor MSRs. The voltage is optimized based on the selected frequency and the number of active processors IA cores.
  - Once the voltage is established, the PLL locks on to the target frequency.
  - All active processor IA cores share the same frequency and voltage. In a multi-core processor, the highest frequency P-state requested among all active IA cores is selected.
  - Software-requested transitions are accepted at any time. If a previous transition is in progress, the new transition is deferred until the previous transition is completed.
- The processor controls voltage ramp rates internally to ensure glitch-free transitions.

---

#### NOTE

Because there is low transition latency between P-states, a significant number of transitions per-second are possible.

---

### 7.4.6 Intel® Turbo Boost Technology 2.0

The Intel® Turbo Boost Technology 2.0 allows the processor IA core/processor graphics core to opportunistically and automatically run faster than the processor IA core base frequency/processor graphics base frequency if it is operating below power,

temperature, and current limits. The Intel® Turbo Boost Technology 2.0 feature is designed to increase the performance of both multi-threaded and single-threaded workloads.

Compared with previous generation products, Intel® Turbo Boost Technology 2.0 will increase the ratio of application power towards Processor Base Power (a.k.a TDP) and also allows to increase power above Processor Base Power (a.k.a TDP) as high as PL2 for short periods of time. Thus, thermal solutions and platform cooling that are designed to less than thermal design guidance might experience thermal and performance issues since more applications will tend to run at the maximum power limit for significant periods of time.

#### 7.4.6.1 Power Monitoring

When operating in turbo mode, the processor monitors its own power and adjusts the processor and graphics frequencies to maintain the average power within limits over a thermally significant time period. The processor estimates the package power for all components on the package. In the event that a workload causes the temperature to exceed program temperature limits, the processor will protect itself using the Adaptive Thermal Monitor.

#### 7.4.6.2 Power Control

Illustration of Intel® Turbo Boost Technology 2.0 power control is shown in the following sections and figures. Multiple controls operate simultaneously allowing customization for multiple systems thermal and power limitations. These controls allow for turbo optimizations within system constraints and are accessible using MSR, MMIO, and PECI interfaces.

#### 7.4.6.3 Frequency

To determine the highest performance frequency amongst active processor IA cores, the processor takes the following into consideration:

- The number of processor IA cores operating in the C0 state.
- The estimated processor IA core current consumption and ICCMax settings.
- The estimated package prior and present power consumption and turbo power limits.
- The package temperature.

Any of these factors can affect the maximum frequency for a given workload. If the power, current, or thermal limit is reached, the processor will automatically reduce the frequency to stay within its TDP limit. Turbo processor frequencies are only active if the operating system is requesting the P0 state. For more information on P-states and C-states, refer to Power Management.

### 7.4.7 Intel® Thermal Velocity Boost

Intel® Thermal Velocity Boost (Intel® TVB) allows the processor IA core to opportunistically and automatically increase the Intel® Turbo Boost Technology 2.0 frequency by up to two speed bins whenever processor temperature allows. The Intel® TVB feature is designed to increase performance of both multi-threaded and single-threaded workloads.



### 7.4.8 Intel® Speed Shift Technology

Intel® Speed Shift Technology is an energy efficient method of frequency control by the hardware rather than relying on OS control. OS is aware of available hardware P-states and requests the desired P-state or it can let the hardware determine the P-state. The OS request is based on its workload requirements and awareness of processor capabilities. Processor decision is based on the different system constraints for example Workload demand, thermal limits while taking into consideration the minimum and maximum levels and activity window of performance requested by the Operating System.

### 7.4.9 Intel® Advanced Vector Extensions 2

Intel® Advanced Vector Extensions 2.0 (Intel® AVX2) is the latest expansion of the Intel instruction set. Intel® AVX2 extends the Intel® Advanced Vector Extensions (Intel® AVX) with 256-bit integer instructions, floating-point fused multiply-add (FMA) instructions, and gather operations. The 256-bit integer vectors benefit math, codec, image, and digital signal processing software. FMA improves performance in face detection, professional imaging, and high-performance computing. Gather operations increase vectorization opportunities for many applications. In addition to the vector extensions, this generation of Intel processors adds new bit manipulation instructions useful in compression, encryption, and general purpose software. For more information on Intel® AVX, refer to <http://www.intel.com/software/avx>

Intel® Advanced Vector Extensions (Intel® AVX) are designed to achieve higher throughput to certain integer and floating point operation. Due to varying processor power characteristics, utilizing AVX instructions may cause a) parts to operate below the base frequency b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software and system configuration and you should consult your system manufacturer for more information.

Intel® Advanced Vector Extensions refers to Intel® AVX or Intel® AVX2.

For more information on Intel® AVX, refer to <https://software.intel.com/en-us/isa-extensions/intel-avx>.

#### 7.4.9.1 Intel® AVX2 Vector Neural Network Instructions

Vector instructions for deep learning extension for AVX2.

### 7.4.10 Intel® 64 Architecture x2APIC

The x2APIC architecture extends the xAPIC architecture that provides key mechanisms for interrupt delivery. This extension is primarily intended to increase processor addressability.

Specifically, x2APIC:

- Retains all key elements of compatibility to the xAPIC architecture:
  - Delivery modes
  - Interrupt and processor priorities
  - Interrupt sources
  - Interrupt destination types

- Provides extensions to scale processor addressability for both the logical and physical destination modes
- Adds new features to enhance the performance of interrupt delivery
- Reduces the complexity of logical destination mode interrupt delivery on link based architectures

The key enhancements provided by the x2APIC architecture over xAPIC are the following:

- Support for two modes of operation to provide backward compatibility and extensibility for future platform innovations:
  - In xAPIC compatibility mode, APIC registers are accessed through memory mapped interface to a 4K-Byte page, identical to the xAPIC architecture.
  - In the x2APIC mode, APIC registers are accessed through the Model Specific Register (MSR) interfaces. In this mode, the x2APIC architecture provides significantly increased processor addressability and some enhancements on interrupt delivery.
- Increased range of processor addressability in x2APIC mode:
  - Physical xAPIC ID field increases from 8 bits to 32 bits, allowing for interrupt processor addressability up to 4G-1 processors in physical destination mode. A processor implementation of x2APIC architecture can support fewer than 32-bits in a software transparent fashion.
  - Logical xAPIC ID field increases from 8 bits to 32 bits. The 32-bit logical x2APIC ID is partitioned into two sub-fields – a 16-bit cluster ID and a 16-bit logical ID within the cluster. Consequently,  $((2^{20}) - 16)$  processors can be addressed in logical destination mode. Processor implementations can support fewer than 16 bits in the cluster ID sub-field and logical ID sub-field in a software agnostic fashion.
- More efficient MSR interface to access APIC registers:
  - To enhance inter-processor and self-directed interrupt delivery as well as the ability to virtualize the local APIC, the APIC register set can be accessed only through MSR-based interfaces in x2APIC mode. The Memory Mapped IO (MMIO) interface used by xAPIC is not supported in x2APIC mode.
- The semantics for accessing APIC registers have been revised to simplify the programming of frequently-used APIC registers by system software. Specifically, the software semantics for using the Interrupt Command Register (ICR) and End Of Interrupt (EOI) registers have been modified to allow for more efficient delivery and dispatching of interrupts.
- The x2APIC extensions are made available to system software by enabling the local x2APIC unit in the “x2APIC” mode. To benefit from x2APIC capabilities, a new operating system and a new BIOS are both needed, with special support for the x2APIC mode.
- The x2APIC architecture provides backward compatibility to the xAPIC architecture and forwards extensible for future Intel platform innovations.

For more information, refer to the Intel® 64 Architecture x2APIC Specification at <http://www.intel.com/products/processor/manuals/>

### 7.4.11 Intel® Dynamic Tuning Technology (DTT)

Intel Dynamic Tuning consists of a set of software drivers and applications that allow a system manufacturer to optimize system performance and usability by:

- Dynamically optimize turbo settings of IA processors, power and thermal states of the platform for optimal performance
- Dynamically adjust the processor's peak power based on the current power delivery capability for optimal system usability
- Dynamically mitigate radio frequency interference for better RF throughput.

### 7.4.12 Intel® GNA 3.0

GNA stands for Gaussian Mixture Model and Neural Network Accelerator.

The GNA is used to process speech recognition without user training sequence. The GNA is designed to unload the processor cores and the system memory with complex speech recognition tasks and improve the speech recognition accuracy. The GNA is designed to compute millions of Gaussian probability density functions per second without loading the processor cores while maintaining low power consumption.

### 7.4.13 Cache Line Write Back (CLWB)

Writes back to memory the cache line (if dirty) that contains the linear address specified with the memory operand from any level of the cache hierarchy in the cache coherence domain. The line may be retained in the cache hierarchy in the non-modified state. Retaining the line in the cache hierarchy is a performance optimization (treated as a hint by hardware) to reduce the possibility of a cache miss on a subsequent access. Hardware may choose to retain the line at any of the levels in the cache hierarchy, and in some cases, may invalidate the line from the cache hierarchy. The source operand is a byte memory location.

The CLWB instruction is documented in the Intel® Architecture Instruction Set Extensions Programming Reference (future architectures):

<https://software.intel.com/sites/default/files/managed/b4/3a/319433-024.pdf>

### 7.4.14 Remote Action Request (RAR)

RAR enables a significant speed up of several inter-processor operations by moving such operations from software (OS or application) to hardware.

The main feature is the speedup of TLB shutdowns.

A single RAR operation can invalidate multiple memory pages in the TLB.

A TLB (Translation Lookaside Buffer) is a per-core cache that holds mappings from virtual to physical addresses.

A TLB shutdown is the process of propagating a change in memory mapping (page table entry) to all the cores.

RAR supports the following operations:

- **Page Invalidation:** imitates the operation of performing INVLPG instructions corresponding or the TLB invalidation corresponding with "MOV CR3 / CR0"

- **Page Invalidation without CR3 Match:** identical to “Page invalidation”, except that the processor does not check for a CR3 match
- **PCID Invalidation:** imitates the operation of performing INVPCID instructions
- **EPT Invalidation:** imitates the operation of performing INVEPT instructions
- **VPID Invalidation:** imitates the operation of performing INVVPID instructions
- **MSR Write:** imitates the operation of WRMSR instructions on all cores

### 7.4.15 User Mode Wait Instructions

The *UMONITOR* and *UMWAIT* are user mode (Ring 3) instructions similar to the supervisor mode (Ring 0) *MONITOR/MWAIT* instructions without the C-state management capability.

*TPAUSE* is an enhanced *PAUSE* instruction.

The mnemonics for the three new instructions are:

- **UMONITOR:** operates just like *MONITOR* but allowed in all rings.
- **UMWAIT:** allowed in all rings, and no specification of target C-state.
- **TPAUSE:** similar to *PAUSE* but with a software-specified delay. Commonly used in spin loops.

## 7.5 Debug Technologies

### 7.5.1 Intel® Processor Trace

Intel® Processor Trace (Intel® PT) is a tracing capability added to Intel® Architecture, for use in software debug and profiling. Intel® PT provides the capability for more precise software control flow and timing information, with limited impact on software execution. This provides an enhanced ability to debug software crashes, hangs, or other anomalies, as well as responsiveness and short-duration performance issues.

Intel® VTune™ Amplifier for Systems and the Intel® System Debugger are part of Intel® System Studio 2015 (and newer) product, which includes updates for the new debug and trace features, including Intel® PT and Intel® Trace Hub.

Intel® System Studio is available for download at <https://software.intel.com/en-us/system-studio>.

An update to the Linux\* performance utility, with support for Intel® PT, is available for download at [https://github.com/virtuoso/linux-perf/tree/intel\\_pt](https://github.com/virtuoso/linux-perf/tree/intel_pt). It requires rebuilding the kernel\* and the perf utility.

### 7.5.2 Platform CrashLog

- The CrashLog feature is intended for use by system builders (OEMs) as a means to triage and perform first level debug of failures.
- CrashLog enables the BIOS or the OS to collect data on failures with the intent to collect and classify the data as well as analyze failure trends.
- CrashLog is a mechanism to collect debug information into a single location and then allow access to that data via multiple methods, including the BIOS and OS of the failing system.

- CrashLog is initiated by a Crash Data Detector on observation of error conditions (TCO watchdog timeout, machine check exceptions, etc.).
- Crash Data Detector notifies the Crash Data Requester of the error condition in order for the Crash Data Requester to collect Crash Data from several different IPs and/or Crash Nodes and stores the data to the Crash Data Storage (on-die SRAM) prior to the reset.
- After the system has rebooted, the Crash Data Collector reads the Crash Data from the Crash Data Storage and makes the data available to either to software and/or back to a central server to track error frequency and trends.

### 7.5.3 Telemetry Aggregator

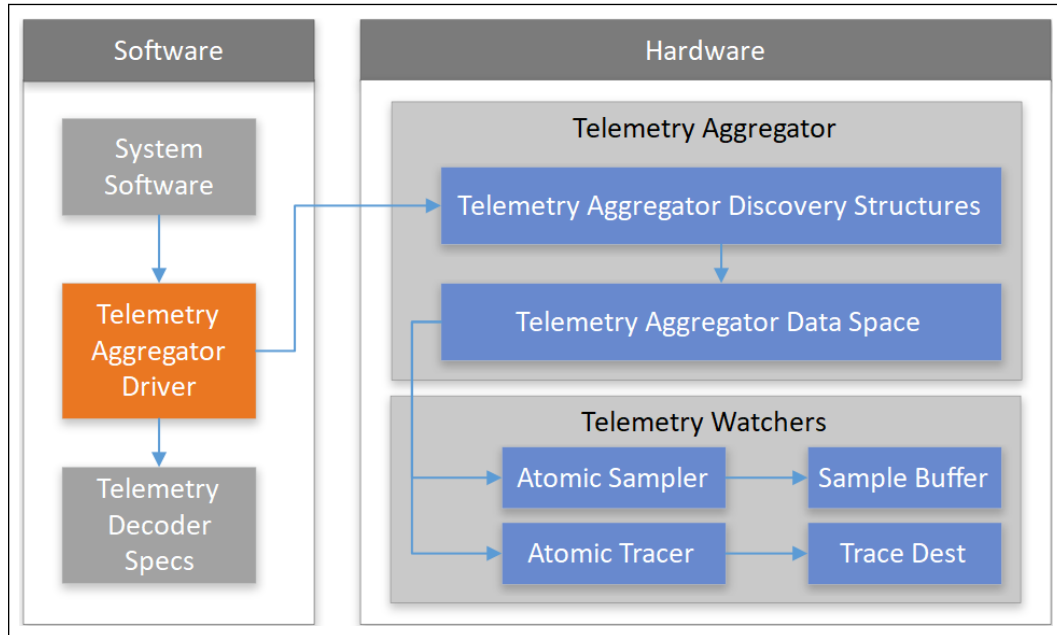
The Telemetry Aggregator serves as an architectural and discoverable interface to hardware telemetry:

- Standardized PCIe discovery solution that enables software to discover and manage telemetry across products
- Standardized definitions for telemetry decode, including data type definitions
- Exposure of commonly used telemetry for power and performance debug including:
  - P-State status, residency and counters
  - C-State status, residency and counters
  - Energy monitoring
  - Device state monitoring (for example, PCIe L1)
  - Interconnect/bus bandwidth counters
  - Thermal monitoring

Exposure of Processor state snapshot for atomic monitoring of package power states, uninterrupted by software that reads.

The Telemetry Aggregator is also a companion to the CrashLog feature where data is captured about the Processor at the point of a crash. These counters can provide insights into the nature of the crash.

**Figure 7. Telemetry Aggregator**



## 8.0 Audio Voice and Speech

**Table 17. Acronyms**

Acronyms	Description
DMA	Direct Memory Access.
DMIC	Digital Microphone. PDM based MEMs microphone modules.
DSP	Digital Signal Processor. In AVS specifically a DSP to process audio data.
I <sup>2</sup> S	Inter IC Sound. A serial bus using PCM.
MEMs	Micro electrical mechanical Systems. For AVS devices such as Digital MEMs Microphones.
MSI	Message Signaled Interrupt. An in-band method of signaling an interrupt.
PCM	Pulse Code Modulation. Modulation with amplitude coded into stream.
PDM	Pulse Density Modulation. Modulation with amplitude coded by pulse density.
SDI	Serial Data In.
SDO	Serial Data Out.
SoC	System On Chip.
VOIP	Voice Over Internet Protocol

### 8.1 Feature Overview

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

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

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

- Intel® High Definition Audio (Intel® HD Audio) Controller Capabilities
- Audio DSP Capabilities
- Intel® High Definition Audio Interface Capabilities
- Direct Attached Digital Microphone (PDM) Interface
- USB Audio Offload Support
- I<sup>2</sup>S / PCM Interface

- Intel® Display Audio Interface
- MIPI\* SoundWire\* Interface

### 8.1.1 Intel® High Definition Audio (Intel® HD Audio) Controller Capabilities

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

- Baseline Intel® HD Audio operation with legacy DMA transporting audio stream to / from audio codecs, with host CPU carrying out the audio processing
- Low power audio operation with offload DMA transporting audio stream to / from the Audio DSP offload engine offloading the audio processing from host CPU
- Ability transport audio stream to various audio codecs speaking different link protocols with the same audio host controller view from SW stacks
- PCI / PCI Express\* controller
- Supports data transfers, descriptor fetches, and DMA position writes using VCO
- Independent Bus Initiator logic for 16 general purpose DMA streams
- Supports variable length stream slots
- Each audio stream supports up to:
  - 16 channels per stream
  - 32 bits/sample
  - 192 kHz sample rate
- Supports memory-based command/response transport
- Supports optional Immediate Command/Response mechanism
- Supports output and input stream synchronization
- Supports global time synchronization
- Supports MSI interrupt delivery
- Support for ACPI D3 and D0 Device States
- Supports Function Level Reset (FLR)
- Support Converged Platform Power Management (CPPM)
  - Support 1 ms of buffering with all DMA running with maximum bandwidth.
  - Support 10 ms of buffering with 1 output DMA and 1 input DMA running at 2 channels, 96 kHz, 16 bit audio.

### 8.1.2 Audio DSP Capabilities

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



- Audio DSP based on 4 Cadence\* Tensilica\* LX6 HIFI3 DSP Cores operating up to 400 MHz with 3 MB SRAM
- Low power support for Intel® Wake on Voice (Intel® WOV)
- Low power audio playback with post processing
- Low power VoIP voice call with pre-processing
- Various DSP functions provided by DSP Core: MP3, AAC, 3rd Party IP Algorithms, etc.
- Host downloadable DSP FW functions
- Voice call processing enhancement
- Sensors algorithm offload / assistance: motion, proximity, etc.
- Supports 4 DSP Cores at 400 MHz and 3 MB SRAM.

### 8.1.3 Intel® High Definition Audio Interface Capabilities

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

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

### 8.1.4 Direct Attached Digital Microphone (PDM) Interface

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

- Four DMIC PDM interfaces with each interface capable of supporting up to 2 digital MEMs microphones.
- Low power always listening support for Intel® Wake on Voice
- 2 PCM audio streams (with independent PCM sampling rate: 48 kHz or 16 kHz) per digital mic interface
- Ultrasound reception capable with higher frequency ranges between 3.84 MHz - 4.8 MHz.
- Support of 1.8 V I/O voltages

### 8.1.5 USB Audio Offload Support

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

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

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

The I<sup>2</sup>S / PCM interface is an optional feature offering connection to the I<sup>2</sup>S / PCM audio codecs. The I<sup>2</sup>S / PCM audio codecs are widely adopted in the phone and tablet platforms as they are typically customized for low power application. The codec structure is typically unique per codec vendor implementation and requires vendor specific SW module for controlling the codec. These I<sup>2</sup>S / PCM audio codecs will be enumerated based on ACPI table or OS specific static configuration information. The Audio DSP is required to be enabled in order to enable I<sup>2</sup>S / PCM link as registers are only addressable through the Audio DSP and its FW. I<sup>2</sup>S/PCM Interface capabilities are listed as follows:

- Up to 3 I<sup>2</sup>S/PCM ports to support multiple I<sup>2</sup>S connections
- Can support 2 modes: Target Mode or Initiator Mode.
- I<sup>2</sup>S audio playback at 2 ch x 192 kHz x 24 bits
- I<sup>2</sup>S audio capture at 2 ch x 192 kHz x 24 bits
- PCM audio playback at 8 ch x 48 kHz x 24 bits
- PCM audio capture at 8 ch x 48 kHz x 24 bits
- Support 5G / 4G modem codec

- Support BT codec HFP / HSP SCO at 8 / 16 kHz
- Support BT codec A2DP at 48 kHz
- Support FM radio codec
- Supports 1.8 V I/O voltages

### 8.1.7 Intel® Display Audio Interface

The Intel® Display Audio codec provides audio stream routing to the integrated HDMI and DP links through the existing Intel® HD Audio controller SW stacks. The Intel® Display Audio codec is enumerated by the Intel® HD Audio driver software if discovered over the Intel Display Audio Interface.

### 8.1.8 MIPI® SoundWire\* Interface

The SoundWire interface is an optional feature offering connection to the SoundWire devices, which include audio codecs and modem codecs. The SoundWire interface is the latest audio interface targeting (but not limited to) the phone and tablet market and the main advantage is the connection simplicity with a two wires multi-drop topology + PCM/PDM streaming capabilities. Currently SoundWire devices are non-standard across different vendors (similar to I<sup>2</sup>S / PCM audio codecs), hence it is very likely to require customized audio codec SW per vendor. These devices will be enumerated based on vendor / device ID of the SoundWire device reporting. SoundWire interface capabilities are listed as follows:

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

## 8.2 Signal Description

Table 18. Signal Descriptions

Name	Type	Description
<b>Intel High Definition Audio Signals</b>		
GPP_R4 / <b>HDA_RST#</b> / I2S2_SCLK / DMIC_CLK_A_0A	O	<b>Intel HD Audio Reset:</b> Initiator H/W reset to internal/external codecs.
GPP_R1 / <b>HDA_SYNC</b> / I2S0_SFRM / DMIC_CLK_B_1A	O	<b>Intel HD Audio Sync:</b> 48 kHz fixed rate frame sync to the codecs. Also used to encode the stream number.
GPP_R0 / <b>HDA_BCLK</b> / I2S0_SCLK / DMIC_CLK_B_0A / HDA_PROC_BCLK	O	<b>Intel HD Audio Bit Clock:</b> Up to 24 MHz serial data clock generated by the Intel HD Audio controller.
GPP_R2 / <b>HDA_SDO</b> / I2S0_TXD / HDA_PROC_SDO	O	<b>Intel HD Audio Serial Data Out:</b> Serial TDM data output to the codecs. The serial output is double-pumped for a bit rate of up to 48 Mb/s.
GPP_R3 / <b>HDA_SDI0</b> / I2S0_RXD / HDA_PROC_SDI	I	<b>Intel HD Audio Serial Data In 0:</b> 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.
GPP_R5 / <b>HDA_SDI1</b> / I2S2_SFRM / DMIC_DATA_0A	I	<b>Intel HD Audio Serial Data In 1:</b> 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.
<b>Intel Display Audio Interface</b>		
GPP_R0 / HDA_BCLK / I2S0_SCLK / DMIC_CLK_B_0A / <b>HDA_PROC_BCLK</b>	O	<b>Display Audio Bit Clock:</b> Serial data clock generated by the Intel HD Audio controller. PCH supports data rate of up to 96 Mb/s.
GPP_R2 / HDA_SDO / I2S0_TXD / <b>HDA_PROC_SDO</b>	O	<b>Display Audio Serial Data Out:</b> Serial TDM data output to the codec. PCH supports data rate of up to 96 Mb/s.
GPP_R3 / HDA_SDI0 / I2S0_RXD / <b>HDA_PROC_SDI</b>	I	<b>Display Audio Serial Data In:</b> Serial TDM data input from the codec. PCH supports data rate of up to 96 Mb/s.
<b>I<sup>2</sup>S/PCM Interface</b>		
GPP_R0 / HDA_BCLK / <b>I2S0_SCLK</b> / DMIC_CLK_B_0A / HDA_PROC_BCLK	I/O	<b>I<sup>2</sup>S/PCM serial bit clock 0:</b> Clock used to control the timing of a transfer. Can be generated internally (Initiator mode) or taken from an external source (Target mode).
GPP_S0 / SNDW0_CLK / <b>I2S1_SCLK</b>	I/O	<b>I<sup>2</sup>S/PCM serial bit clock 1:</b> This clock is used to control the timing of a transfer. Can be generated internally (Initiator mode) or taken from an external source (Target mode).
GPP_R4 / HDA_RST# / <b>I2S2_SCLK</b> / DMIC_CLK_A_0A	I/O	<b>I<sup>2</sup>S/PCM serial bit clock 2:</b> This clock is used to control the timing of a transfer. Can be generated internally (Initiator mode) or taken from an external source (Target mode).
GPP_R1 / HDA_SYNC / <b>I2S0_SFRM</b> / DMIC_CLK_B_1A	I/O	<b>I<sup>2</sup>S/PCM serial frame indicator 0:</b> This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Initiator mode) or taken from an external source (Target mode).
<i>continued...</i>		

Name	Type	Description
GPP_S1 / SNDW0_DATA / <b>I2S1_SFRM</b>	I/O	<b>I<sup>2</sup>S/PCM serial frame indicator 1:</b> This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Initiator mode) or taken from an external source (Target mode).
GPP_R5 / HDA_SDI1 / <b>I2S2_SFRM</b> / DMIC_DATA_0A	I/O	<b>I<sup>2</sup>S/PCM serial frame indicator 2:</b> This signal indicates the beginning and the end of a serialized data word. Can be generated internally (Initiator mode) or taken from an external source (Target mode).
GPP_R2 / HDA_SDO / <b>I2S0_TXD</b> / HDA_PROC_SDO	O	<b>I<sup>2</sup>S/PCM transmit data (serial data out)0:</b> This signal transmits serialized data. The sample length is a function of the selected serial data sample size.
GPP_S2 / SNDW1_CLK / DMIC_CLK_A_0 / <b>I2S1_TXD</b>	O	<b>I<sup>2</sup>S/PCM transmit data (serial data out)1:</b> This signal transmits serialized data. The sample length is a function of the selected serial data sample size.
GPP_R6 / <b>I2S2_TXD</b> / DMIC_CLK_1A	O	<b>I<sup>2</sup>S/PCM transmit data (serial data out)2:</b> This signal transmits serialized data. The sample length is a function of the selected serial data sample size.
GPP_R3 / HDA_SDI0 / <b>I2S0_RXD</b> / HDA_PROC_SDI	I	<b>I<sup>2</sup>S/PCM receive data (serial data in)0:</b> This signal receives serialized data. The sample length is a function of the selected serial data sample size.
GPP_S3 / SNDW1_DATA / DMIC_DATA_0 / <b>I2S1_RXD</b>	I	<b>I<sup>2</sup>S/PCM receive data (serial data in)1:</b> This signal receives serialized data. The sample length is a function of the selected serial data sample size.
GPP_R7 / <b>I2S2_RXD</b> / DMIC_DATA_1A	I	<b>I<sup>2</sup>S/PCM receive data (serial data in)2:</b> This signal receives serialized data. The sample length is a function of the selected serial data sample size.
GPP_D19 / <b>I2S_MCLK1_OUT</b>	O	<b>I<sup>2</sup>S/PCM Initiator reference clock 1:</b> This signal is the initiator reference clock that connects to an audio codec.
<b>DMIC Interface</b>		
GPP_S2 / SNDW1_CLK / <b>DMIC_CLK_A_0</b> / I2S1_TXD or GPP_R4 / HDA_RST# / I2S2_SCLK / <b>DMIC_CLK_A_0A</b>	O	<b>Digital Mic Clock A0:</b> Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz. May be duplicated into CLKA and CLKB for individual left / right DMIC power control.
GPP_S6 / SNDW3_CLK / <b>DMIC_CLK_A_1</b> or GPP_R6 / I2S2_TXD / <b>DMIC_CLK_1A</b>	O	<b>Digital Mic Clock A1:</b> Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz. May be duplicated into CLKA and CLKB for individual left / right DMIC power control.
GPP_S4 / SNDW2_CLK / <b>DMIC_CLK_B_0</b> or GPP_R0 / HDA_BCLK / I2S0_SCLK / <b>DMIC_CLK_B_0A</b> / HDA_PROC_BCLK	O	<b>Digital Mic Clock B0:</b> Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz. May be duplicated into CLKA and CLKB for individual left / right DMIC power control.
GPP_S5 / SNDW2_DATA / DMIC_CLKB1 or GPP_R1 / HDA_SYNC / I2S0_SFRM / <b>DMIC_CLK_B_1</b>	O	<b>Digital Mic Clock B1:</b> Serial data clock generated by the HD Audio controller. The clock output frequency is up to 4.8 MHz. May be duplicated into CLKA and CLKB for individual left / right DMIC power control.
GPP_S3 / SNDW1_DATA / <b>DMIC_DATA_0</b> / I2S1_RXD or	I	<b>Digital Mic Data:</b> Serial data input from the digital mic.
<i>continued...</i>		

Name	Type	Description
GPP_R5 / HDA_SDI1 / I2S2_SFRM / <b>DMIC_DATA_0A</b>		
GPP_S7 / SNDW3_DATA / <b>DMIC_DATA_1</b> or GPP_R7 / I2S2_RXD / <b>DMIC_DATA_1A</b>	I	<b>Digital Mic Data:</b> Serial data input from the digital mic.
<b>SoundWire Interface</b>		
GPP_S0 / <b>SNDW0_CLK</b> / I2S1_SCLK	I/O	<b>SoundWire Clock:</b> Serial data clock to external peripheral devices.
GPP_S1 / <b>SNDW0_DATA</b> / I2S1_SFRM	I/O	<b>SoundWire Data:</b> Serial data input from external peripheral devices.
GPP_S2 / <b>SNDW1_CLK</b> / DMIC_CLK_A_0 / I2S1_TXD	I/O	<b>SoundWire Clock:</b> Serial data clock to external peripheral devices.
GPP_S3 / <b>SNDW1_DATA</b> / DMIC_DATA_0 / I2S1_RXD	I/O	<b>SoundWire Data:</b> Serial data input from external peripheral devices.
GPP_S4 / <b>SNDW2_CLK</b> / DMIC_CLK_B_0	I/O	<b>SoundWire Clock:</b> Serial data clock to external peripheral devices.
GPP_S5 / <b>SNDW2_DATA</b> / DMIC_CLK_B_1	I/O	<b>SoundWire Data:</b> Serial data input from external peripheral devices.
GPP_S6 / <b>SNDW3_CLK</b> / DMIC_CLK_A_1	I/O	<b>SoundWire Clock:</b> Serial data clock to external peripheral devices.
GPP_S7 / <b>SNDW3_DATA</b> / DMIC_DATA_1	I/O	<b>SoundWire Data:</b> Serial data input from external peripheral devices.
<b>SNDW_RCOMP</b>	I/O	<b>SoundWire RCOMP:</b> 200ohm +/- 1% compensation resistor required to ground.
<b>Misc</b>		
GPP_B14 / <b>SPKR</b> / TIME_SYNC1 / SATA_LED# / ISH_GP6	O	<b>Speaker Output:</b> Used for connection to external speaker for POST sounds if not using HD_Audio embedded option.

### 8.3 Integrated Pull-Ups and Pull-Downs

Table 19. Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value ( Ω )
HDA_SYNC	Pull-down	20 kohm
HDA_SDO	Pull-down	20 kohm
HDA_SDI[1:0]	Pull-down	20 kohm
HDACPU_SDO	Pull-down	20 kohm
HDACPU_SDI	Pull-down	20 kohm
I2S[2:0]_SCLK	Pull-down	20 kohm
I2S[2:0]_SFRM	Pull-down	20 kohm
I2S[2:0]_TXD	Pull-down	20 kohm
I2S[2:0]_RXD	Pull-down	20 kohm
<i>continued...</i>		

Signal	Resistor Type	Value ( $\Omega$ )
I2S_MCLK	Pull-down	20 kohm
I2S_MCLK1_OUT	Pull-down	20 kohm
DMIC_DATA[1:0]	Pull-down	20 kohm
SNDW[3:0]_DATA	Pull-down	5 kohm
SPKR	Pull-down	20 kohm

## 8.4 I/O Signal Planes and States

**Table 20. I/O Signal Planes and States**

Signal Name	Power Plane	During Reset <sup>2</sup>	Immediately After Reset <sup>2</sup>	S3/S4/S5	Deep Sx
<b>High Definition Audio Interface</b>					
HDA_RST#	Primary	Driven Low	Driven Low	Driven Low	OFF
HDA_SYNC	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
HDA_BCLK	Primary	Driven Low	Driven Low	Driven Low	OFF
HDA_SDO	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
HDA_SDI[1:0]	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
<b>I<sup>2</sup>S/PCM Interface</b>					
I2S[2:0]_SCLK	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S[2:0]_SFRM	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S[2:0]_TXD	Primary	Driven Low	Driven Low	Driven Low	OFF
I2S[2:0]_RXD	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
I2S_MCLK1_OUT	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
<b>DMIC Interface</b>					
DMIC_CLKA[1:0]	Primary	Driven Low	Driven Low	Driven Low	OFF
DMIC_CLKB[1:0]	Primary	Driven Low	Driven Low	Driven Low	OFF
DMIC_DATA[1:0]	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
<b>SoundWire Interface</b>					
SNDW[3:0]_DATA	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SNDW[3:0]_CLK	Primary	Driven Low	Driven Low	Driven Low	OFF
<b>Misc</b>					
SPKR	Primary	Internal Pull-down	Driven Low	Low then disabled (refer to note)	OFF
<p>Notes: 1. SPKR and I2S0_TXD are also straps in which the pull-down only occurs during the sampling window and then the pull-ups are disabled.</p> <p>2. Reset reference for primary well pins is RSMRST#.</p>					

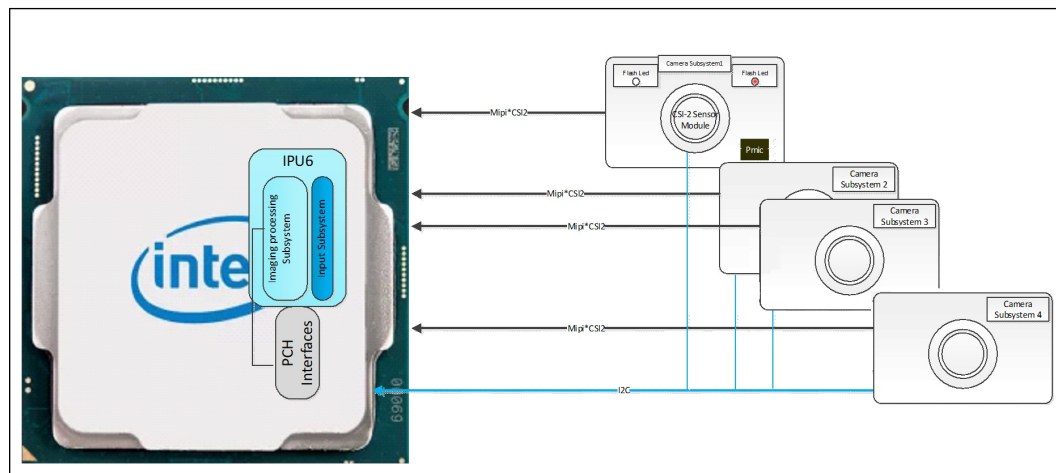
## 9.0 Image Processing Unit

### 9.1 Platform Imaging Infrastructure

The platform imaging infrastructure is based on the following hardware components:

- **Camera Subsystem:** Located in the lid of the system and contains CMOS sensor, flash, LED, I/O interface (MIPI\* CSI-2 and I2C\*), focus control and other components.
- **Camera I/O Controller:** The I/O controller is located in the processor and contains a MIPI-CSI2 host controller. The host controller is a PCI device (independent of the IPU device). The CSI-2 HCI brings imaging data from an external image into the system and provides a command and control channel for the image using I<sup>2</sup>C.
- **Intel® IPU (Image Processing Unit):** The IPU processes raw images captured by Bayer sensors. The result images are used by still photography and video capture applications (JPEG, H.264, and so on.).

Figure 8. Processor Camera System



### 9.2 Intel® Image Processing Unit (Intel® IPU6)

IPU6 is Intel's 6th generation solution for an Imaging Processing Unit, providing advanced imaging functionality for Intel processors, as well as more specialized functionality for High Performance Mobile Phones, Automotive, Digital Surveillance Systems (DSS), and other market segments.

IPU6 is a continuing evolution of the architecture introduced in IPU4 and enhanced in IPU5. Additional image quality improvements are introduced, as well as hardware accelerated support for temporal de-noising and new sensor technologies such as Spatially Variant Exposure HDR and Dual Photo Diode, among others.



IPU6 provides a complete high quality hardware accelerated pipeline, and is therefore not dependent on algorithms running on the vector processors to provide the highest quality output.

## 9.3 Camera/MIPI

### 9.3.1 Camera Pipe Support

The IPU6 fixed function pipe supports the following functions:

- Black level correction;
- White balance;
- Color matching;
- Lens shading (vignette) correction;
- Color crosstalk (color shading) correction;
- Dynamic defect pixel replacement;
- Auto-focus-pixel (PDAF) hiding;
- High quality demosaic;
- Scaling and format conversion;
- Temporal noise reduction running on Intel graphics.

### 9.3.2 MIPI\* CSI-2 Camera Interconnect

The Camera I/O Controller provides a native/integrated interconnect to camera sensors, compliant with MIPI\* CSI-2 V2.0 protocol. Total of 8 data+4 clock lanes are available for the camera interface supporting up to 4 sensors .

Data transmission interface (referred as CSI-2) is a unidirectional differential serial interface with data and clock signals; the physical layer of this interface is the MIPI\* Alliance Specification for D-PHY.

The control interface (referred as CCI) is a bi-directional control interface compatible with I<sup>2</sup>C standard.

#### 9.3.2.1 Camera Control Logic

The camera infrastructure supports several architectural options for camera control utilizing camera PMIC and/or discrete logic. IPU6 control options utilize I<sup>2</sup>C for bidirectional communication and PCH GPIOs to drive various control functions.

#### 9.3.2.2 Camera Modules

Intel maintains an Intel User Facing Camera Approved Vendor List and Intel World-Facing Approved Vendor List to simplify system design. Additional services are available to support non-AVL options.

### 9.3.2.3 MIPI\* CSI-2 Interface Signals

Signal Name	Description	Dir.	Buffer Type	Link Type
CSI_A_DP[1:0] CSI_A_DN[1:0]	CSI-2 Ports Data lane	I	DPHY	Diff
CSI_A_CLK_P CSI_A_CLK_N	CSI 2 Port A Clock lane	I	DPHY	Diff
CSI_B_DP[3:0] CSI_B_DN[3:0]	CSI-2 Ports Data lane	I	DPHY	Diff
CSI_B_CLK_P CSI_B_CLK_N	CSI 2 Port B Clock lane	I	DPHY	Diff
CSI_C_DP[3:0] CSI_C_DN[3:0]	CSI-2 Ports Data lane	I	DPHY	Diff
CSI_C_CLK_P CSI_C_CLK_N	CSI 2 Port C Clock lane	I	DPHY	Diff
CSI_D_DP[1:0] CSI_D_DN[1:0]	CSI-2 Ports Data lane	I	DPHY	Diff
CSI_D_CLK_P CSI_D_CLK_N	CSI 2 Port D Clock lane	I	DPHY	Diff
CSI_RCOMP	CSI Resistance Compensation	N/A	N/A	SE

### 9.3.2.4 CSI-2 Lane Configuration

Table 21. CSI-2 Lane Configuration

Port Data/Clock	Configuration Option 1	Configuration Option 2
Port A Clock	NA	x2
Port A Lane 0	x4	
Port A Lane 1		
Port B Clock		x4
Port B Lane 0		
Port B Lane 1		
Port C Clock	x4	x2
Port C Lane 0		
Port C Lane 1		
Port D Lane 0	x4	x2
Port D Lane 1		
Port D Clock		

## 10.0 Power Management

---



---

**NOTE**

In this chapter, Sx refers to S3/S4/S5 states; Deep Sx refers to Deep S4/Deep S5 states.

---

**Table 22. Acronyms**

Acronyms	Description
PMC	Power Management Controller
STD	Suspend To Disk
VR	Voltage Regulator

**Table 23. References**

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

Figure 9. Processor Power States

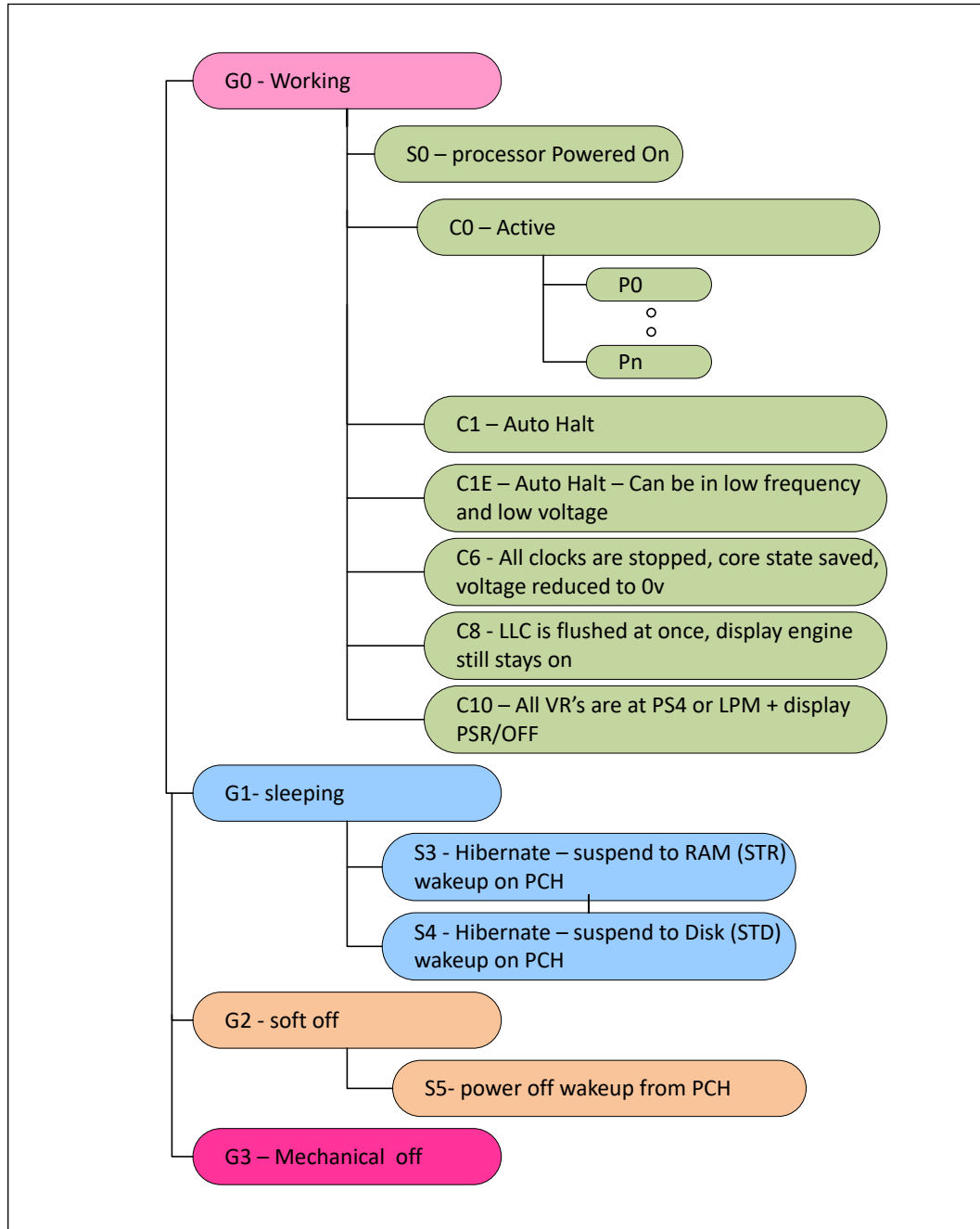
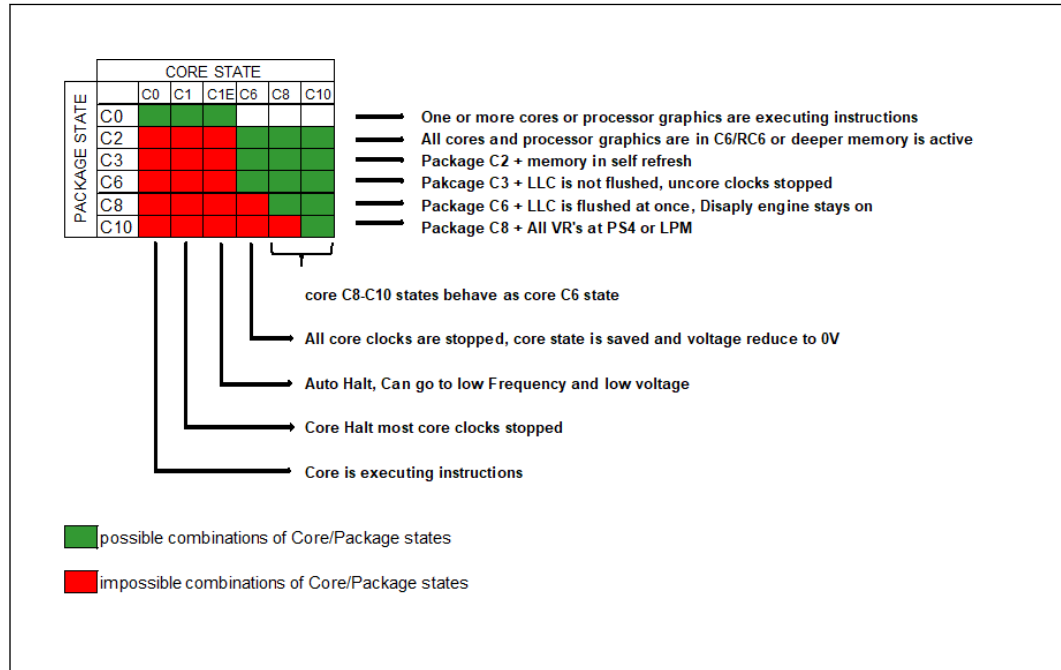


Figure 10. Processor Package and IA Core C-States



1. PkgC2/C3 are non-architectural: software cannot request to enter these states explicitly. These states are intermediate states between PkgC0 and PkgC6.
2. There are constraints that prevent the system to go deeper.
3. The "core state" relates to the core which is in the HIGEST power state in the package (most active).

### 10.1 Signal Description

Name	Type	Description
GPD1 / <b>ACPRESENT</b>	I	<b>ACPRESENT:</b> This input pin indicates when the platform is plugged into AC power or not. In addition to Intel® CSE 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. <i>Note:</i> An external pull-up resistor is required.
GPD0 / <b>BATLOW#</b>	I	<b>Battery Low:</b> An input from the battery to indicate that there is insufficient power to boot the system. Assertion will prevent wake from S3/S4/S5 states or exit from Deep Sx state. This signal can also be enabled to cause an SMI# when asserted. This signal is multiplexed with GPD0. <i>Note:</i> For any platform not using this pin functionality, this signal must be tied high to VCCDSW_3P3. An external pull-up resistor to VCCDSW_3P3 is required.
GPP_B0 / <b>CORE_VID0</b>	O	<b>PCH Core VID Bit 0:</b> May connect to discrete VR on platform. In default mode this pin is driven high ('1').
GPP_B1 / <b>CORE_VID1</b>	O	<b>PCH Core VID Bit 1:</b> May connect to discrete VR on platform. In default mode this pin is driven high ('1').

*continued...*

Name	Type	Description
GPP_H18 / <b>PROC_C10_GATE#</b>	O	<b>External Power Gate:</b> Control for VCCIO, VCCSTG and VCCPLL_OC during C10. When asserted, VCCIO, VCCSTG and VCCPLL_OC can be 0 V, however the power good indicators for these rails must remain asserted. <i>Note:</i> An external pull-up resistor to the DRAM power plane is required.
<b>DSW_PWROK</b>	I	<b>DeepSx Well PWROK:</b> Power OK Indication for the VCCDSW_3p3 voltage rail. <i>Note:</i> This signal is in the RTC well. This signal cannot tie with RSMRST#.
<b>PCH_PWROK</b>	I	<b>PCH Power OK:</b> When asserted, PCH_PWROK is an indication to the PCH that all of its core power rails have been stable. The platform may drive asynchronously. When PCH_PWROK is de-asserted, the PCH asserts PLTRST#. <i>Notes:</i> <ul style="list-style-type: none"> <li>PCH_PWROK must not glitch, even if RSMRST# is low</li> <li>An external pull-down resistor is required.</li> </ul>
GPP_B13 / <b>PLTRST#</b>	O	<b>Platform Reset:</b> The PCH asserts PLTRST# to reset devices on the platform. The PCH asserts PLTRST# low in Sx states and when a cold, warm, or global reset occurs. The PCH de-asserts PLTRST# upon exit from Sx states and the aforementioned resets. There is no guaranteed minimum assertion time for PLTRST#.
GPP_B11 / <b>PMCALERT#</b>	I/OD	<b>PMC Alert Pin:</b> Supports USB-C* PD controller architecture.
GPD3 / <b>PWRBTN#</b>	I	<b>Power Button:</b> The Power Button may cause an SMI# or SCI to indicate a system request to go to a sleep state. If the system is already in a sleep state, this signal will cause a wake event. If PWRBTN# is pressed for more than 4 seconds (default; timing is configurable), this will cause an unconditional transition (power button override) to the S5 state. Override will occur even if the system is in the S3-S4 states. This signal has an internal Pull-up resistor and has an internal 16 ms de-bounce on the input. <i>Note:</i> Upon entry to S5 due to a power button override, if Deep Sx is enabled and conditions are met, the system will transition to Deep S5.
<b>RSMRST#</b>	I	<b>Primary Well Reset:</b> This signal is used for resetting the primary power plane logic. This signal must be asserted for at least 10 ms after the primary power wells are valid. When de-asserted, this signal is an indication that the primary power wells are stable. <i>Note:</i> An external pull down resistor is required
GPD6 / <b>SLP_A#</b>	O	<b>SLP_A#:</b> Signal asserted when the Intel® CSE platform goes to M-Off. If you are not using SLP_A# for any functional purposes on your platform, or can tolerate lack of minimum assertion time, program the "SLP_A# minimum assertion width" value to the minimum. SLP_A# functionality can be utilized on the platform via either the physical pin or via the SLP_A# virtual wire over eSPI. <i>Note:</i> An external pull down resistor is required
GPD9 / <b>SLP_WLAN#</b>	O	<b>WLAN Sub-System Sleep Control:</b> When SLP_WLAN# is asserted, power can be shut off to the external wireless LAN device. SLP_WLAN# will always be de-asserted in S0. If you are not using SLP_WLAN# for any functional purposes on your platform, or can tolerate lack of minimum assertion time, program the "SLP_A# minimum assertion width" value to the minimum.
GPP_B12 / <b>SLP_S0#</b>	O	<b>S0 Sleep Control:</b> When PCH is idle and processor is in C10 state, this pin will assert to indicate VR controller can go into a light load mode. This signal can also be connected to EC for other power management related optimizations. <i>Note:</i> An external pull-up resistor is required.
GPD4 / <b>SLP_S3#</b>	O	<b>S3 Sleep Control:</b> SLP_S3# is for power plane control. This signal shuts off power to all non-critical systems when in the S3, S4, or S5 state. <i>Note:</i> An external pull-down resistor is required.
GPD5 / <b>SLP_S4#</b>	O	<b>S4 Sleep Control:</b> SLP_S4# is for power plane control. This signal shuts power to all non-critical systems when in the S4 or S5 state. <i>Notes:</i> <ul style="list-style-type: none"> <li>This pin must be used to control the DRAM power in order to use the PCH DRAM power-cycling feature.</li> <li>An external pull-down resistor is required.</li> </ul>

*continued...*

Name	Type	Description
GPD10 / <b>SLP_S5#</b>	O	<b>S5 Sleep Control:</b> SLP_S5# is for power plane control. This signal is used to shut power off to all non-critical systems when in the S5 state. <i>Note:</i> An external pull-down resistor is required.
<b>SLP_SUS#</b>	O	<b>Deep Sx Indication:</b> When asserted (driven low), this signal indicates PCH is in Deep Sx state where internal primary power is shut off for enhanced power saving. When de-asserted (driven high), this signal indicates exit from Deep Sx state and primary power can be applied to PCH. For non- Deep Sx, this pin also needs to use to turn on VCCPRIM_1P8 VR. This pin cannot left unconnected. <i>Notes:</i> <ul style="list-style-type: none"> <li>This is in the DSW power well</li> <li>An external pull-down resistor is required.</li> </ul>
<b>SPIVCCIOSEL</b>	I	<b>SPI Operation Voltage Select</b> There is no internal pull-up or pull-down on the strap. An external resistor is required. 0 = SPI voltage is 3.3 V (4.7 kohm pull-down to GND), 1 = SPI voltage is 1.8V (4.7 kohm pull-up to VCCDSW_3p3).
GPP_A3 / ESPI_IO3 / <b>SUSACK#</b>	I	<b>SUSACK#:</b> If Deep Sx is supported, the EC/motherboard controlling logic must change SUSACK# to match SUSWARN# once the EC/motherboard controlling logic has completed the preparations discussed in the description for the SUSWARN# pin. <i>Note:</i> SUSACK# is only required to change in response to SUSWARN# if Deep Sx is supported by the platform.
GPD8 / <b>SUSCLK</b>	O	<b>Suspend Clock:</b> This clock is a digitally buffered version of the RTC clock.
GPP_A2 / ESPI_IO2 / <b>SUSWARN#</b> / SUSPWRDNACK	O	<b>SUSWARN#:</b> This pin asserts low when the PCH is planning to enter the Deep Sx power state and remove Primary power (using SLP_SUS#). The EC/motherboard controlling logic must observe edges on this pin, preparing for primary well power loss on a falling edge and preparing for Primary well related activity (host/Intel CSE wakes and runtime events) on a rising edge. SUSACK# must be driven to match SUSWARN# once the above preparation is complete. SUSACK# should be asserted within a minimal amount of time from SUSWARN# assertion as no wake events are supported if SUSWARN# is asserted but SUSACK# is not asserted. Platforms supporting Deep Sx, but not wishing to participate in the handshake during wake and Deep Sx entry may tie SUSACK# to SUSWARN#. This pin is multiplexed with SUSPWRDNACK since it is not needed in Deep Sx supported platforms.
GPP_A2 / ESPI_IO2 / SUSWARN# / <b>SUSPWRDNACK</b>	O	<b>SUSPWRDNACK:</b> Active high. Asserted by the PCH on behalf of the Intel CSE when it does not require the PCH Primary well to be powered. Platforms are not expected to use this signal when the PCH Deep Sx feature is used.
GPP_H3 / <b>SX_EXIT_HOLDOFF#</b>	I	<b>Sx Exit Holdoff Delay:</b> Delay exit from Sx state after SLP_A# is de-asserted. <i>Note:</i> When eSPI is enabled, SX_EXIT_HOLDOFF# functionality is not available, and assertion of the signal will not impact Sx exit flows.
<b>SYS_PWROK</b>	I	<b>System Power OK:</b> This generic power good input to the PCH is driven and utilized in a platform-specific manner. While PCH_PWROK always indicates that the 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. <i>Note:</i> An external pull-down resistor is required
<b>SYS_RESET#</b>	I	<b>System Reset:</b> This pin forces an internal reset after being de-bounced. <i>Note:</i> An external pull-up resistor is required.
GPP_B15 / <b>TIME_SYNC0</b> / ISH_GP7	I	<b>Time Synchronization:</b> Used for synchronization both input (latch time when pin asserted) and output (toggle pin when programmed time is hit).
GPP_B14/ <b>TIME_SYNC1</b> / SPKR / SATA_LED# / ISH_GP6	I	<b>Time Synchronization:</b> Used for synchronization both input (latch time when pin asserted) and output (toggle pin when programmed time is hit).
GPP_B2 / <b>VRALERT#</b>	I	<b>VR Alert:</b> ICC Max. throttling indicator from the PCH voltage regulators. VRALERT# pin allows the VR to force PCH throttling to prevent an over current shutdown. PMC based on the VRALERT# and messages from the processor. The messages from the processor allows the processor to constrain the PCH to a particular power budget.

*continued...*

Name	Type	Description
<b>WAKE#</b>	I/OD	<b>PCI Express* Wake Event in Sx:</b> Input Pin in Sx. Sideband wake signal on PCI Express* asserted by components requesting wake up. <i>Notes:</i> <ul style="list-style-type: none"> <li>This is an output pin during S0ix states hence this pin can not be used to wake up the system during S0ix states.</li> <li>An external pull-up resistor is required.</li> </ul>
<b>VCCST_OVERRIDE</b>	O	<b>VccST Override:</b> Signal that allows the PCH to keep VCCST powered ON (in case VCCST is powered down) for USB-C wake capability (connected to VCCSTPWGOOD_TCSS on board). Signal will stay high when plug-in device on USB Type-C Subsystem port and signal will stay low when no device is connected.
GPP_F22 / <b>VNN_CTRL</b>		<b>VNN_Control:</b> External bypass rail control pin. Without requiring BIOS to be involved during the S0ix states. This pin use to control of the VCC_VNNEXT_1P05 voltage.
GPP_F23 / <b>V1P05_CTRL</b>	Out	<b>V1p05_Control:</b> External bypass rail control pin. Without requiring BIOS to be involved during the S0ix states. This pin use to control of the VCC_V1P05EXT_1P05 voltage

## 10.2 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
<b>ACPRESENT</b>	Pull-down	15 kohm - 40 kohm	1
<b>PWRBTN#</b>	Pull-up	20 kohm +/- 30%	
<b>SUSACK#</b>	Pull-up	20 kohm +/- 30%	
<b>WAKE#</b>	Pull-down	15 kohm - 40 kohm	1

*Note:* 1. Pull-down is configurable and can be enabled in Deep Sx state; refer to DSX\_CFG register for more details.

## 10.3 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>18</sup>	Immediately after Reset <sup>18</sup>	S3/S4/S5	Deep Sx
<b>ACPRESENT<sup>6,10,15</sup></b>	DSW	Undriven /Driven Low <sup>4</sup>	Undriven	Undriven	Undriven/Internal Pull-down <sup>8</sup>
<b>BATLOW#</b>	DSW	Undriven	Undriven	Undriven	OFF
<b>CORE_VID0<sup>11,17</sup></b>	Primary	Driven High	Driven High	Driven High	OFF
<b>CORE_VID1<sup>11,17</sup></b>	Primary	Driven High	Driven High	Driven High	OFF
<b>CPU_C10_GATE#<sup>1,17</sup></b>	Primary	Undriven <sup>19</sup>	Undriven <sup>19</sup>	Driven Low	OFF
<b>DRAM_RESET#<sup>14</sup></b>	DSW	Undriven	Undriven	Undriven	Undriven
<b>DSW_PWROK</b>	RTC	Undriven	Undriven	Undriven	Undriven
<b>SPIVCCIOSEL</b>	DSW	Undriven	Undriven	Undriven	Undriven
<b>PCH_PWROK</b>	RTC	Undriven	Undriven	Undriven	Undriven
<b>PLTRST#<sup>16</sup></b>	Primary	Driven Low	Driven High	Driven Low	OFF
<b>PWRBTN#<sup>15</sup></b>	DSW	Internal Pull-up	Internal Pull-up	Internal Pull-up	Internal Pull-up
<b>RSMRST#</b>	RTC	Undriven	Undriven	Undriven	Undriven

**continued...**



Signal Name	Power Plane	During Reset <sup>18</sup>	Immediately after Reset <sup>18</sup>	S3/S4/S5	Deep Sx
SLP_A# <sup>6,16</sup>	DSW	Driven Low	Driven High	Driven High/ Driven Low <sup>12</sup>	Driven High/ Driven Low <sup>12</sup>
SLP_S0# <sup>1</sup>	Primary	Driven High	Driven High	Driven High	OFF
SLP_S3# <sup>6,16</sup>	DSW	Driven Low	Driven High	Driven Low	Driven Low
SLP_S4# <sup>6,16</sup>	DSW	Driven Low	Driven High	Driven Low	Driven Low <sup>9</sup>
SLP_S5# <sup>6,16</sup>	DSW	Driven Low	Driven High	Driven High/ Driven Low <sup>3</sup>	Driven High/ Driven Low <sup>9</sup>
SLP_SUS# <sup>6,14</sup>	DSW	Driven Low	Driven High	Driven High	Driven Low
SLP_WLAN# <sup>6,16</sup>	DSW	Driven Low	Driven Low	Driven High/ Driven Low <sup>7</sup>	Driven High/ Driven Low <sup>7</sup>
SUSACK# <sup>15</sup>	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up	OFF
SUSCLK <sup>10,16</sup>	DSW	Driven Low	Toggling	Toggling	Toggling <sup>10</sup>
SUSWARN# / SUSWRDNACK <sup>10,16</sup>	Primary	Driven Low	Driven Low	Driven Low <sup>5</sup>	OFF
SX_EXIT_HOLDOFF# <sup>15</sup>	Primary	Undriven	Undriven	Undriven	OFF
SYS_PWROK	Primary	Undriven	Undriven	Undriven	OFF
SYS_RESET#	Primary	Undriven	Undriven	Undriven	OFF
VREALERT# <sup>15</sup>	Primary	Undriven	Undriven	Undriven	OFF
WAKE# <sup>13</sup>	DSW	Undriven	Undriven	Undriven	Undriven/Internal Pull-down

Notes:

1. Driven High during S0 and driven Low during S0i3 when all criteria for assertion are met.
2. SLP\_S4# is driven 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® CSE state. SUSWRDNACK is always '0' while in M0, but can be driven to '0' or '1' while in M0ff state. SUSWRDNACK is the default mode of operation. If Deep Sx is supported, then subsequent boots will default to SUSWRDNACK.
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 CSE state.
8. Pull-down is configurable and can be enabled in Deep Sx state; refer to DSX\_CFG register for more details.
9. When platform enters Deep Sx, the SLP\_S4# and SLP\_S5# pin will retain the value it held prior to Deep Sx entry.
10. Internal weak pull-down resistor is enabled during power sequencing.
11. The CORE\_VID pins defaults to '1' and will be driven to '1' to reflect that voltage will support 1.8 V. The VID able to change to 1.8 V/ 3.3 V based on the CPU and the state.
12. Pin state is a function of whether the platform is configured to have Intel CSE on or off in Sx.
13. Output High-Z, not glitch free.
14. Output High-Z, glitch free with ~1 k Pull-down during respective power sequencing
15. Output High-Z, not glitch free.
16. Output High-Z, glitch free with ~20 k Pull-down during respective power sequencing.
17. Output High-Z, glitch free with ~20 k Pull-up during respective power sequencing.
18. Reset reference for primary well pins is RSMRST#, DSW well pins is DSW\_PWROK, and RTC well pins is RTRCRST#.
19. Sx can be optionally be high when RSMRST# is high and the buffer moves to its native mode at which point it will become low.

## 10.4 Functional Description

This section provides information on the following topics:

- Features
- PCH S0 Low Power
- Power Management Sub-state
- PCH and System Power States
- SMI#/SCI Generation
- C-States
- Sleep States
- Event Input Signals and Their Usage
- ALT Access Mode
- System Power Supplies, Planes, and Signals
- Reset Behavior

### 10.4.1 Features

- Support for *Advanced Configuration and Power Interface (ACPI)* providing power and thermal management
  - ACPI 24-Bit Timer SCI and SMI# Generation
- PCI PME# signal for Wake Up from Low-Power states
- System Sleep State Control
  - ACPI S3 state – Suspend to RAM (STR)
  - ACPI S4 state – Suspend-to-Disk (STD)
  - ACPI G2/S5 state – Soft Off (SOFF)
  - Power Failure Detection and Recovery
  - Deep Sx
- Intel® CSE Power Management Support
  - Wake events from the Intel® CSE (enabled from all S-States including Catastrophic S5 conditions)
- SLP\_S0# signal for external platform VR power gating or EC power management handling during lower power conditions.

### 10.4.2 PCH S0 Low Power

The PCH has many independent functions and I/O interfaces making power management a highly distributive task. The first level of power management is to control the independent resources and the best place to do that is in the controllers. The second level of power management is to control the shared resources, which requires communication amongst the users of the shared resources.

The PCH power states are a combination of first level and second level power management functions. The **deeper** the power state, meaning the lower power required, generally means that more resources are disabled.

### 38.4 MHz Crystal Shutdown

When the CPU and system are in a power management state that can tolerate gating the 38.4 MHz crystal clock, this circuit can be powered down. This occurs when the processor enters C10 state, and all other consumers of the 38.4 MHz XTAL de-assert their clock request.

### SLP\_S0#

SLP\_S0# is the indication to the system to enter the deterministic idle state (S0i3). This is a PCH hardware controlled output pin. This signal is defined as active low which means a 0 V indicates the deterministic idle state. Additional power saving steps such as VPCLVM may happen during this state.

## 10.4.3 Power Management Sub-state

### S0ix State Enable

If a platform wants to disable certain S0ix states, BIOS can do so by modifying the LPM\_EN register. The mapping of S0ix states to bits in the LPM\_EN register are given below:

**Table 24. LPM\_EN Register Mapping**

Bit Number	S0ix State	Required Implementation <sup>1</sup>
0	S0i2.0	None <sup>2</sup>
1	S0i3.0	None <sup>2</sup>

### NOTES

1. Other board capabilities such as power control for RTD3 cold may be implicitly required to satisfy requirements.
2. For external bypass voltage selection, VNN\_CTRL can be used to select the external bypass.

## 10.4.4 PCH and System Power States

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

**Table 25. General Power States for Systems Using the PCH**

State / Substates	Legacy Name/Description
G0/S0/C0	<b>Full On:</b> Processor operating. Individual devices may be shut down or be placed into lower power states to save power.
G0/S0/Cx	<b>Cx States:</b> C states are processor power states within the S0 system state that provide for various levels of power savings on the processor. The processor manages C states itself. The actual C state is not passed to the PCH. Only C state related messages are sent to the PCH and PCH will base its behavior on the actual data passed.
<i>continued...</i>	

State / Substates	Legacy Name/Description
G1/S3	<b>Suspend-To-RAM (STR):</b> The system context is maintained in system DRAM, but power is shut off to non-critical circuits. Memory is retained and refreshes continue. All external clocks stop except RTC.
G1/S4	<b>Suspend-To-Disk (STD):</b> The context of the system is maintained on the disk. All power is then shut off to the system except for the logic required to resume.
G2/S5	<b>Soft Off (SOFF):</b> System context is not maintained. All power is shut off except for the logic required to restart. A full boot is required when waking.
S0ix	<b>S0 Idle States:</b> Processor PKG C states and platform latency tolerance will allow the PCH to decide when to take aggressive power management actions.
Deep Sx	<b>Deep Sx:</b> An optional low power state where system context may or may not be maintained depending upon entry condition. All power is shut off except for minimal logic that allows exiting Deep Sx. If Deep Sx state was entered from S4 state, then the resume path will place system back into S4. If Deep Sx state was entered from S5 state, then the resume path will place system back into S5.
G3	<b>Mechanical OFF (M-Off):</b> System context not maintained. All power is shut off except for the RTC. No "Wake" events are possible. This state occurs if the user removes the main system batteries in a mobile system, turns off a mechanical switch, or if the system power supply is at a level that is insufficient to power the "waking" logic. When system power returns, transition will depend on the state just prior to the entry to G3 and the AFTERG3_EN bit in the General Power Management Configuration (GEN_PMCON). Refer to table <a href="#">System Power Plane</a> for more details.

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

**NOTE**

Transitions among the various states may appear to temporarily transition through intermediate states. For example, in going from S0 to S5, it may appear to pass through the G1/S3/S4 state. These intermediate transitions and states are not listed in the table below.

**Table 26. State Transition Rules for the PCH**

Present State	Transition Trigger	Next State
G0/S0/C0	<ul style="list-style-type: none"> <li>• SLP_EN bit set</li> <li>• Power Button Override<sup>3,5</sup></li> <li>• Mechanical Off/Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>• G0/S0/Cx</li> <li>• G1/S3, G1/S4, or G2/S5 state</li> <li>• G2/S5</li> <li>• G3</li> </ul>
G0/S0/Cx	<ul style="list-style-type: none"> <li>• Power Button Override<sup>3,5</sup></li> <li>• Mechanical Off/Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>• G0/S0/C0</li> <li>• S5</li> <li>• G3</li> </ul>
G1/S3	<ul style="list-style-type: none"> <li>• Any Enabled Wake Event</li> <li>• Power Button Override<sup>3,5</sup></li> <li>• Mechanical Off/Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>• G0/S0/C0<sup>2</sup></li> <li>• G2/S5</li> <li>• G3</li> </ul>
G1/S4	<ul style="list-style-type: none"> <li>• Any Enabled Wake Event</li> <li>• Power Button Override<sup>3,5</sup></li> <li>• Conditions met as described in <a href="#">PCH and System Power States</a> on page 91</li> <li>• Mechanical Off/Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>• G0/S0/C0<sup>2</sup></li> <li>• G2/S5</li> <li>• Deep S4</li> <li>• G3</li> </ul>
G2/S5	<ul style="list-style-type: none"> <li>• Any Enabled Wake Event</li> </ul>	<ul style="list-style-type: none"> <li>• G0/S0/C0<sup>2</sup></li> </ul>

*continued...*

Present State	Transition Trigger	Next State
	<ul style="list-style-type: none"> <li>Conditions met as described in <a href="#">PCH and System Power States</a> on page 91</li> <li>Mechanical Off/Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>Deep S5</li> <li>G3</li> </ul>
G2/Deep Sx	<ul style="list-style-type: none"> <li>Any Enabled Wake Event</li> <li>ACPRESENT Assertion</li> <li>Mechanical Off/Power Failure</li> <li>Power Button Override</li> </ul>	<ul style="list-style-type: none"> <li>G0/S0/C0<sup>2</sup></li> <li>G1/S4 or G2/S5 (Refer to <a href="#">PCH and System Power States</a> on page 91)</li> <li>G3</li> <li>G2/S5</li> </ul>
G3	<ul style="list-style-type: none"> <li>Power Returns</li> </ul>	<ul style="list-style-type: none"> <li>S0/C0 (reboot) or G2/S5<sup>4</sup> (stay off until power button pressed or other wake event)<sup>1,2</sup></li> </ul>
<p><i>Notes:</i> 1. Some wake events can be preserved through power failure.                  2. Transitions from the S3–S4–S5 states to the S0 state are deferred until BATLOW# is inactive.                  3. Includes all other applicable types of events that force the host into and stay in G2/S5.                  4. If the system was in G1/S4 before G3 entry, then the system will go to S0/C0 or G1/S4.                  5. Upon entry to S5 due to a power button override, if Deep S5 is enabled and conditions are met per section <a href="#">PCH and System Power States</a> on page 91, the system will transition to Deep S5.</p>		

### System Power Planes

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

#### NOTE

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

**Table 27. System Power Plane**

Plane	Controlled By	Description
Processor	SLP_S3# signal	The SLP_S3# signal can be used to cut the power to the processor completely.
Main (Applicable to Platform, PCH does not have a Main well)	SLP_S3# signal	<p>When SLP_S3# goes active, power can be shut off to any circuit not required to wake the system from the S3 state. Since the S3 state requires that the memory context be preserved, power must be retained to the main memory.</p> <p>The processor, PCI Express* will typically be power-gated when the Main power plane is shut down, although there may be small subsections powered.</p> <p><i>Note:</i> The PCH power is not controlled by the SLP_S3# signal, but instead by the SLP_SUS# signal.</p>
Memory	SLP_S4# signal SLP_S5# signal	<p>When SLP_S4# goes active, power can be shut off to any circuit not required to wake the system from the S4. Since the memory context does not need to be preserved in the S4 state, the power to the memory can also be shut down.</p> <p>When SLP_S5# goes active, power can be shut off to any circuit not required to wake the system from the S5 state. Since the memory context does not need to be preserved in the S5 state, the power to the memory can also be shut down.</p>
<i>continued...</i>		

Plane	Controlled By	Description
Intel® CSE	SLP_A#	SLP_A# signal is asserted when the Intel® CSE goes to M-Off.
Primary Well	SLP_SUS#	This signal is asserted when the Primary 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.

### 10.4.5 SMI#/SCI Generation

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

Once the SMI VLW has been delivered, the PCH takes no action on behalf of active SMI events until Host software sets the End of SMI (EOS) bit. At that point, if any SMI events are still active, the PCH will send another SMI VLW message.

The SCI is a level-mode interrupt that is typically handled by an ACPI-aware operating system. In non-APIC systems (which is the default), the SCI IRQ is routed to one of the 8259 interrupts (IRQ 9, 10, or 11). The 8259 interrupt controller must be programmed to level mode for that interrupt.

In systems using the APIC, the SCI can be routed to interrupts 9, 10, 11, 20, 21, 22, or 23. The interrupt polarity changes depending on whether it is on an interrupt shareable with a PIRQ or not. The interrupt remains asserted until all SCI sources are removed.

The table below shows which events can cause an SMI and SCI.

**NOTE**

Some events can be programmed to cause either an SMI or SCI. The usage of the event for SCI (instead of SMI) is typically associated with an ACPI-based system. Each SMI or SCI source has a corresponding enable and status bit.

**Table 28. Causes of SMI and SCI**

Cause	SCI	SMI	Additional Enables <sup>1</sup>	Where Reported
PME#	Yes	Yes	PME_EN=1	PME_STS
PME_B0 (Internal, Bus 0, PME-Capable Agents)	Yes	Yes	PME_B0_EN=1	PME_B0_STS
PCI Express* PME Messages	Yes	Yes	PCI_EXP_EN=1 (Not enabled for SMI)	PCI_EXP_STS
PCI Express* Hot-Plug Message	Yes	Yes	HOT_PLUG_EN=1 (Not enabled for SMI)	HOT_PLUG_STS
Power Button Press	Yes	Yes	PWRBTN_EN=1	PWRBTN_STS
Power Button Override (Note 6)	Yes	No	None	PWRBTNOR_STS
RTC Alarm	Yes	Yes	RTC_EN=1	RTC_STS
ACPI Timer overflow (2.34 seconds)	Yes	Yes	TMROF_EN=1	TMROF_STS

*continued...*

Cause	SCI	SMI	Additional Enables <sup>1</sup>	Where Reported
GPIO	Yes	Yes	Refer to Note 8	
TCO SCI message from processor	Yes	No	None	CPUSCI_STS
TCO SCI Logic	Yes	No	TCOSCI_EN=1	TCOSCI_STS
TCO SMI Logic	No	Yes	TCO_EN=1	TCO_STS
TCO SMI – Year 2000 Rollover	No	Yes	None	NEWCENTURY_STS
TCO SMI – TCO TIMEROOUT	No	Yes	None	TIMEOUT
TCO SMI – OS writes to TCO_DAT_IN register	No	Yes	None	OS_TCO_SMI
TCO SMI – NMI occurred (and NMIs mapped to SMI)	No	Yes	NMI2SMI_EN=1	TCO_STS, NMI2SMI_STS
TCO SMI – INTRUDER# signal goes active	No	Yes	INTRD_SEL=10	INTRD_DET
TCO SMI – Changes of the WPD (Write Protect Disable) bit from 0 to 1	No	Yes	LE (Lock Enable)=1	BIOSWR_STS
TCO SMI – Write attempted to BIOS	No	Yes	WPD=0	BIOSWR_STS
BIOS_RLS written to 1 (Note 7)	Yes	No	GBL_EN=1	GBL_STS
GBL_RLS written to	No	Yes	BIOS_EN=1	BIOS_STS
Write to B2h register	No	Yes	APMC_EN = 1	APM_STS
Periodic timer expires	No	Yes	PERIODIC_EN=1	PERIODIC_STS
64 ms timer expires	No	Yes	SWSMI_TMR_EN=1	SWSMI_TMR_STS
Enhanced USB Legacy Support Event	No	Yes	LEGACY_USB2_EN = 1	LEGACY_USB2_STS
Serial IRQ SMI reported	No	Yes	None	SERIRQ_SMI_STS
Device monitors match address in its range	No	Yes	Refer to DEVTRAP_STS register description	DEVTRAP_STS
SMBus Host Controller	No	Yes	SMB_SMI_EN, Host Controller Enabled	SMBus host status reg.
SMBus Target SMI message	No	Yes	None	SMBUS_SMI_STS
SMBus SMBALERT# signal active	No	Yes	None	SMBUS_SMI_STS
SMBus Host Notify message received	No	Yes	HOST_NOTIFY_INTREN	SMBUS_SMI_STS, HOST_NOTIFY_STS
BATLOW# assertion	Yes	Yes	BATLOW_EN=1	BATLOW_STS
Access microcontroller 62h/66h	No	Yes	MCSMI_EN	MCSMI_STS
SLP_EN bit written to 1	No	Yes	SMI_ON_SLP_EN=1	SMI_ON_SLP_EN_STS
SPI Command Completed	No	Yes	None	SPI_SMI_STS
eSPI SCI/SMI Request <sup>9</sup>	Yes	Yes	eSPI_SCI_EN	eSPI_SCI_STS eSPI_SMI_STS
Software Generated GPE	Yes	Yes	SWGPE_EN=1	SWGPE_STS
Intel® CSE	Yes	Yes	CSE_SCI_EN=1 CSE_SCI_EN=0; CSE_SMI_EN=1;	CSE_SCI_STS CSE_SMI_STS

*continued...*

Cause	SCI	SMI	Additional Enables <sup>1</sup>	Where Reported
GPIO Lockdown Enable bit changes from '1' to '0'	No	Yes	GPIO_UNLOCK_SMI_EN=1	GPIO_UNLOCK_SMI_STS
USB 3.2 (xHCI) SMI Event	No	Yes	xHCI_SMI_EN=1	xHCI_SMI_STS
Wake Alarm Device Timer	Yes	Yes	WADT_EN	WADT_STS
ISH	Yes	No	ISH_EN	ISH_STS
RTC update-in-progress	No	Yes	Refer to Vol2	RTC_UIP_SMI_STS
SIO SMI events	No	Yes	SIO_SMI_EN	SIO_SMI_STS
SCC	No	Yes	SCC_SMI_EN	SCC_SMI_STS

**Notes:**

1. SCI\_EN must be 1 to enable SCI, except for BIOS\_RLS. SCI\_EN must be 0 to enable SMI.
2. SCI can be routed to cause interrupt 9:11 or 20:23 (20:23 only available in APIC mode).
3. GBL\_SMI\_EN must be 1 to enable SMI.
4. EOS must be written to 1 to re-enable SMI for the next 1.
5. The PCH must have SMI fully enabled when the PCH is also enabled to trap cycles. If SMI is not enabled in conjunction with the trap enabling, then hardware behavior is undefined.
6. When a power button override first occurs, the system will transition immediately to S5. The SCI will only occur after the next wake to S0 if the residual status bit (PRBTNOR\_STS) is not cleared prior to setting SCI\_EN.
7. GBL\_STS being set will cause an SCI, even if the SCI\_EN bit is not set. Software must take great care not to set the BIOS\_RLS bit (which causes GBL\_STS to be set) if the SCI handler is not in place.
8. Refer to [General Purpose Input and Output](#) on page 217 for specific GPIOs enabled for SCIs and/or SMIs
9. eSPI target must assert SCI at least 100 us for the SCI event to be recognized.

### PCI Express\* SCI

PCI Express\* ports and the processor have the ability to cause PME using messages. When a PME message is received, the PCH will set the PCI\_EXP\_STS bit. If the PCI\_EXP\_EN bit is also set, the PCH can cause an SCI using the GPE0\_STS (replaced GPE1\_STS) register.

### PCI Express\* Hot-Plug

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

## 10.4.6 C-States

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

## 10.4.7 Sleep States

### Sleep State Overview

The PCH supports different sleep states (S3/S4/S5), which are entered by methods such as setting the SLP\_EN bit or due to a Power Button press. The entry to the Sleep states is based on several assumptions:

- The G3 state cannot be entered using any software mechanism. The G3 state indicates a complete loss of power.



## Initiating Sleep State

Sleep states (S3/S4/S5) are initiated by:

- Masking interrupts, turning off all bus initiator enable bits, setting the desired type in the SLP\_TYP field, and then setting the SLP\_EN bit. The hardware then attempts to gracefully put the system into the corresponding Sleep state.
- Pressing the PWRBTN# Signal for more than 4 seconds to cause a Power Button Override event. In this case the transition to the S5 state is less graceful, since there are no dependencies from the processor or on clocks other than the RTC clock.
- Assertion of the THERMTRIP# signal will cause a transition to the S5 state. This can occur when system is in the S0 state.
- Shutdown by integrated manageability functions (ASF).
- Internal watchdog timer timeout events.

**Table 29. 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 motherboard uses the SLP_S4# signal to shut off the power to the memory subsystem and any other unneeded subsystem. Only devices needed to wake from this state should be powered.
S5	The PCH asserts SLP_S3#, SLP_S4# and SLP_S5#.

## Exiting Sleep States

Sleep states (S3/S4/S5) are exited based on wake events. The wake events forces the system to a full on state (S0), although some non-critical subsystems might still be shut off and have to be brought back manually. For example, the storage subsystem may be shut off during a sleep state and have to be enabled using a GPIO pin before it can be used.

Upon exit from the PCH-controlled Sleep states, the WAK\_STS bit is set. The possible causes of wake events (and their restrictions) are shown in the table below.

### NOTE

If the BATLOW# signal is asserted, the PCH does not attempt to wake from an S3/S4/S5 state, nor will it exit from Deep Sx state, even if the power button is pressed. This prevents the system from waking when the battery power is insufficient to wake the system. Wake events that occur while BATLOW# is asserted are latched by the PCH, and the system wakes after BATLOW# is de-asserted.

**Table 30. Causes of Wake Events**

Cause	How Enabled	Wake from Sx	Wake from Deep Sx	Wake from Sx After Power Loss <sup>2</sup>	Wake from "Reset" Types <sup>3</sup>
RTC Alarm	Set RTC_EN bit in PM1_EN_STS register.	Yes	Yes	Yes	No
Power Button	Always enabled as Wake event.	Yes	Yes	Yes	Yes
Any GPIOs except DSW GPIOs can be enabled for wake	Refer to Note 5	Yes	No	No	No
Intel® High Definition Audio	Event sets PME_B0_STS bit; PM_B0_EN must be enabled. Can not wake from S5 state if it was entered due to power failure or power button override.	Yes	No	Yes	No
Primary PME#	PME_B0_EN bit in GPE0_EN[127:96] register.	Yes	No	Yes	No
Secondary PME#	Set PME_EN bit in GPE0_EN[127:96] register.	Yes	No	Yes	No
PCI Express* WAKE# pin	PCIEXP_WAKE_DIS bit.	Yes	Yes	Yes	No
SMBALERT#	Refer to Note 4	Yes	No	Yes	Yes
SMBus Target Wake Message (01h)	Wake/SMI# command always enabled as a Wake event. <i>Note:</i> SMBus Target Message can wake the system from S3/S4/S5, as well as from S5 due to Power Button Override.	Yes	No	Yes	Yes
SMBus Host Notify message received	HOST_NOTIFY_WKEN bit SMBus Target Command register. Reported in the SMB_WAK_STS bit in the GPE0_STS register.	Yes	No	Yes	Yes
Intel® CSE Non-Maskable Wake	Always enabled as a wake event.	Yes	No	Yes	Yes
Wake Alarm Device	WADT_EN in GPE0_EN[127:96]	Yes	Yes	No	No
<b>continued...</b>					

Cause	How Enabled	Wake from Sx	Wake from Deep Sx	Wake from Sx After Power Loss <sup>2</sup>	Wake from "Reset" Types <sup>3</sup>
AC_PRESENT	ACPRESENT_WAKE_EN (Note 6)	No	Yes	No	No
USB connection in/after Deep Sx	GPE0_EN.USB_CON_DSX_EN+	Refer to Note 7	Yes	No	No

Notes: 1. If BATLOW# signal is low, PCH will not attempt to wake from S3/S4/S5 (nor will it exit Deep Sx), even if a valid wake event occurs. This prevents the system from waking when battery power is insufficient to wake the system. However, once BATLOW# de-asserts, the system will boot.

2. This column represents what the PCH would honor as wake events but there may be enabling dependencies on the device side which are not enabled after a power loss.

3. Reset Types include: Power Button override, Intel® CSE-initiated power button override, Intel® CSE-initiated host partition reset with power down, Intel® CSE Watchdog Timer, SMBus unconditional power down, processor thermal trip, PCH catastrophic temperature event.

4. SMBALERT# signal is multiplexed with a GPIO pin that defaults to GPIO mode. Hence, SMBALERT# related wakes are possible only when this GPIO is configured in native mode, which means that BIOS must program this GPIO to operate in native mode before this wake is possible. Because GPIO configuration is in the resume well, wakes remain possible until one of the following occurs: BIOS changes the pin to GPIO mode, a G3 occurs or Deep Sx entry occurs.

5. There are only 72 bits in the GPE registers to be assigned to GPIOs, though any of the GPIOs can trigger a wake, only those status of GPIO mapped to 1-tier scheme are directly accessible through the GPE status registers. For those GPIO mapped under 2-tier scheme, their status would be reflected under single initiator status, "GPIO\_TIER2\_SCI\_STS" or GPE0\_STS and further comparison needed to know which 2-tier GPI(s) has triggered the GPIO Tier 2 SCI.

6. A change in ACPRESENT causes an exit from Deep Sx to Sx, but the system will not wake all the way to S0.

7. Connection of a USB device can cause a wake from normal Sx as well. But that class of wakes is routed through PME\_B0, not through this wake enable. The USB\_CON\_DSX\_EN applies only to connection wakes while in Deep Sx or while in Sx after Deep Sx. Note: Sx after Deep Sx reached due to an Intel® CSE wake from Deep Sx or due to ACPRESENT going high while in Deep Sx if Deep Sx is only enabled while on DC power. The following additional conditions are required for this wake to occur:

- The bit(s) in PM\_CFG2.USB\_DSX\_PER\_PORT\_EN associated with the port(s) which experienced the connection must be set to '1'.
- DSX\_CFG.USB\_CON\_DSX\_MODE must be set to '1', routing USB connection to generate a wake rather than be reflected out to a pin

### PCI Express\* WAKE# Signal and PME Event Message

PCI Express\* ports can wake the platform from S3, S4, S5, or Deep Sx using the WAKE# pin. WAKE# is treated as a wake event, but does not cause any bits to go active in the GPE\_STS register.

#### NOTE

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

PCI Express\* ports and the processor have the ability to cause PME using messages. These are logically OR'd to set the single PCI\_EXP\_STS bit. When a PME message is received, the PCH will set the PCI\_EXP\_STS bit. If the PCI\_EXP\_EN bit is also set, the PCH can cause an SCI via GPE0\_STS register.

### Sx-G3-Sx, Handling Power Failures

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

The AFTERG3\_EN bit provides the ability to program whether or not the system should boot once power returns after a power loss event. If the policy is to not boot, the system remains in an S5 state (unless previously in S4). There are only three possible events that will wake the system after a power failure.

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

DSW\_PWROK going low would place the PCH into a G3 state.

Although PME\_EN is in the RTC well, this signal cannot wake the system after a power loss. PME\_EN is cleared by RTCRST#, and PME\_STS is cleared by RSMRST#.

**Table 31. Transitions Due to Power Failure**

State at Power Failure	AFTERG3_EN Bit	Transition when Power Returns and BATLOW# is inactive
S0, S3	1 0	S5 S0
S4	1 0	S4 S0
S5	1 0	S5 S0
Deep S4	1 0	Deep S4 S0
Deep S5	1 0	Deep S5 S0
<i>Notes:</i> 1. Entry state to Deep Sx is preserved through G3 allowing resume from Deep Sx to take appropriate path (that is, return to S4 or S5). 2. G3 related Power Failure is defined as DSW_PWROK transition low.		

### Deep Sx

To minimize power consumption while in S4/S5, the PCH supports a lower power, lower featured version of these power states known as Deep Sx. In the Deep Sx state, the primary wells are powered off, while the Deep Sx Well (DSW) remains powered. A limited set of wake events are supported by the logic located in the DSW.

The Deep Sx capability and the SUSPWRDNACK pin functionality are mutually exclusive.

- **Entry Into Deep Sx**

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

- RTCPMCFG.INT\_SUS\_PD\_EN = 1  
 Intel® CSE must program this bit prior to initiating CMOFF entry
- Intel® CSE in CMOFF

- Deep Sx conditions are checked during CMOFF and CM3-PG entry. If Deep Sx entry would have been allowed if the ACPRESENT signal had been high, PMC FW will enable ACPRESENT as an interrupt source, initiating Deep Sx entry if the power source changes to match the required state
- Host in S3, S4, or S5 and combination of S-state and power source matches the host policy bits
  - ((S3 AC\_GATE\_SUS AND S3) OR (S4AC\_GATE\_SUS AND S4) OR S3 (S5AC\_GATE\_SUS AND S5))

OR

- ((ACPRESENT = 0) AND ((S3DC\_GATE\_SUS AND S3) OR (S4DC\_GATE\_SUS AND S4) OR (S5DC\_GATE\_SUS AND S5)))
- Either Deep Sx entry is not determined by BATLOW# state or BATLOW# is asserted
  - REQ\_BATLOW\_DSX == '0' OR BATLOW# == '0'
- Either Deep Sx entry is not determined by connectivity wake enable or connectivity wake is enabled
  - REQ\_CNV\_NOWAKE\_DSX == '0' OR SLP\_WLAN\_VAL == '0'

**Table 32. Supported Deep Sx Policy Configurations**

Configuration	S4DC_GATE_SUS	S4AC_GATE_SUS	S5DC_GATE_SUS	S5AC_GATE_SUS
1. Enabled in S5 Battery Only (ACPRESENT = 0)	0	0	1	0
1. Enabled in S5 (ACPRESENT not considered)	0	0	1	1
1. Enabled in S4 and S5 when on Battery only (ACPRESENT = 0)	1	0	1	0
1. Enabled in S4 and S5 (ACPRESENT not considered)	1	1	1	1
1. Enabled in S3, S4, and S5 when on Battery only (ACPRESENT = 0)	1	0	1	0
1. Enabled in S3, S4, and S5 (ACPRESENT not considered)	1	1	1	1
1. Deep S4 / S5 disabled	0	0	0	0

*Note:* All other configurations are RESERVED.

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

• **Exit from Deep Sx**

While in Deep Sx, the PCH monitors and responds to a limited set of wake events (RTC Alarm, Power Button and WAKE#). Upon sensing an enabled Deep Sx wake event, the PCH brings up the primary well by de-asserting SLP\_SUS#.

**Table 33. Deep Sx Wake Events**

Event	Enable
RTC Alarm	RTC_EN bit in PM1_EN_STS Register
Power Button	Always enabled
PCIe* WAKE# pin	PCIEXP_WAKE_DIS
Wake Alarm Device	WADT_EN in GPE0_EN

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

### 10.4.8 Event Input Signals and Their Usage

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

#### PWRBTN# (Power Button)

The PCH PWRBTN# signal operates as a “Fixed Power Button” as described in the *Advanced Configuration and Power Interface Specification*. PWRBTN# signal has a 16 ms de-bounce on the input. The state transition descriptions are included in the below table.

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

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

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

Table 34. Transitions Due to Power Button

Present State	Event	Transition/Action	Comment
S0/Cx	PWRBTN# goes low	SMI or SCI generated (depending on SCI_EN, PWRBTN_EN and GLB_SMI_EN)	Software typically initiates a Sleep state <i>Note:</i> Processing of transitions starts within 100 us of the PWRBTN# input pin to PCH going low. <sup>1</sup>
S3 – S5	PWRBTN# goes low	Wake Event. Transitions to S0 state	Standard wakeup <i>Note:</i> Could be impacted by SLP_* min assertion. The minimum time the PWRBTN# pin should be asserted is 150 us. The PCH will start processing this change once the minimum time requirement is satisfied. <sup>1</sup>
Deep Sx	PWRBTN# goes low	Wake Event. Transitions to S0 state	Standard wakeup <i>Note:</i> Could be impacted by SLP_* min assertion. The minimum time the PWRBTN# pin should be asserted is 150 us. The PCH will start processing this change once the minimum time requirement is satisfied but subsequently the PWRBTN# pin needs to de-assert for at least 500 us after RSMRST# de-assertion otherwise the system waits indefinitely in S5 state. <sup>1</sup>
G3	PWRBTN# pressed	None	No effect since no power Not latched nor detected <i>Notes:</i> 1. During G3 exit, PWRBTN# pin must be kept de-asserted for a minimum time of 500 us after the RSMRST# has de-asserted. <sup>2</sup> 2. Beyond this point, the minimum time the PWRBTN# pin has to be asserted to be registered by PCH as a valid wake event is 150 us. <sup>1</sup>
S0 – S4	PWRBTN# held low for at least 4 3 consecutive seconds	Unconditional transition to S5 state and if Deep Sx is enabled and conditions are met, the system will then transition to Deep Sx.	No dependence on processor or any other subsystem <i>Note:</i> Due to internal PCH latency, it could take up to an additional ~1.3s after PWRBTN# has been held low for 4s before the system would begin transitioning to S5.
<i>Notes:</i> 1. If PM_CFG.PB_DB_MODE='0', the debounce logic adds 16 ms to the start/minimum time for processing of power button assertions. 2. This minimum time is independent of the PM_CFG.PB_DB_MODE value. 3. The amount of time PWRBTN# must be asserted is configurable via PM_CFG.PBOP. 4 seconds is the default.			

### Power Button Override Function

If PWRBTN# is observed active for at least four consecutive seconds (always sampled after the output from debounce logic), the PCH should unconditionally transition to the G2/S5 state or Deep Sx, regardless of present state (S0 – S4), even if the

PCH\_PWROK is not active. In this case, the transition to the G2/S5 state or Deep Sx does not depend on any particular response from the processor, nor any similar dependency from any other subsystem.

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

The PWRBTN# status is readable to check if the button is currently being pressed or has been released. If PM\_CFG.PB\_DB\_MODE='0', the status is taken after the de-bounce. If PM\_CFG.PB\_DB\_MODE='1', the status is taken before the de-bounce. In either case, the status is readable using the PWRBTN\_LVL bit.

---

**NOTE**

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

---

**Sleep Button**

The *Advanced Configuration and Power Interface Specification* defines an optional Sleep button. It differs from the power button in that it only is a request to go from S0 to S3–S4 (not S5). Also, in an S5 state, the Power Button can wake the system, but the Sleep Button cannot.

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

**PME# (PCI Power Management Event)**

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

There is also an internal PME\_B0\_STS bit that will be set by the PCH when any internal device with PCI Power Management capabilities on bus 0 asserts the equivalent of the PME# signal.

**SYS\_RESET# Signal**

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

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



---

## NOTES

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

## THERMTRIP# Signal

If THERMTRIP# goes active, the processor is indicating an overheat condition, and the PCH immediately transitions to an S5 state, driving SLP\_S3#, SLP\_S4#, SLP\_S5# low, and setting the GEN\_PMCON\_2.PTS bit. The transition will generally look like a power button override.

When a THERMTRIP# event occurs, the PCH will power down immediately without following the normal S0 -> S5 path. The PCH will immediately drive SLP\_S3#, SLP\_S4#, and SLP\_S5# low within 1 us after sampling THERMTRIP# active.

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

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

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

---

## NOTE

A thermal trip event will clear the PWRBTN\_STS bit.

---

## Sx\_Exit\_Holdoff#

When S3/S4/S5 is entered and SLP\_A# is asserted, Sx\_Exit\_Holdoff# can be asserted by a platform component to delay resume to S0. SLP\_A# de-assertion is an indication of the intent to resume to S0, but this will be delayed so long as Sx\_Exit\_Holdoff# is asserted. Sx\_Exit\_Holdoff is ignored outside of an S3/S4/S5 entry sequence with SLP\_A# asserted. With the de-assertion of RSMRST# (either from G3->S0 or DeepSx->S0), this pin is a GPIO input and must be programmed by BIOS to operate as Sx\_Exit\_Holdoff. When SLP\_A# is asserted (or it is de-asserted but Sx\_Exit\_Holdoff# is asserted), the PCH will not access SPI Flash. How a platform uses this signal is platform specific.

### Requirements to support Sx\_Exit\_Holdoff#

If the PCH is in G3/DeepSx or in the process of exiting G3/DeepSx (RSMRST# is asserted), the EC must not allow RSMRST# to de-assert until the EC completed its flash accesses.

After the PCH has booted up to S0 at least once since the last G3 or DeepSx exit, the EC can begin monitoring SLP\_A# and using the SX\_EXIT\_HOLDOFF# pin to stop the PCH from accessing flash. When SLP\_A# asserts, if the EC intends to access flash, it will assert SX\_EXIT\_HOLDOFF#. To cover the case where the PCH is going through a global reset, and not a graceful Sx+CMoff/Sx+CM3PG entry, the EC must monitor the SPI flash CS0# pin for 5 ms after SLP\_A# assertion before making the determination that it is safe to access flash.

- If no flash activity is seen within this 5 ms window, the EC can begin accessing flash. Once its flash accesses are complete, the EC de-asserts (drives to '1') SX\_EXIT\_HOLDOFF# to allow the PCH to access flash.
- If flash activity is seen within this 5 ms window, the PCH has gone through a global reset. And so the EC must wait until the PCH reaches S0 again before re-attempting the holdoff flow.

---

#### NOTE

When eSPI is enabled, SX\_EXIT\_HOLDOFF# functionality is not available, and assertion of the signal will not impact Sx exit flows.

---

## 10.4.9 ALT Access Mode

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

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

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

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

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

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

### Write Only Registers with Read Paths in ALT Access Mode

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

**Table 35. Write Only Registers with Read Paths in ALT Access Mode**

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

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

### PIC Reserved Bits

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

**Table 36. PIC Reserved Bits Return Values**

PIC Reserved Bits	Value Returned
ICW2(2:0)	000
ICW4(7:5)	000
<i>continued...</i>	

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

## 10.4.10 System Power Supplies, Planes, and Signals

### Power Plane Control

The SLP\_S3# output signal can be used to cut power to the system core supply, since it only goes active for the Suspend-to-RAM state (typically mapped to ACPI S3). Power must be maintained to the PCH primary well, and to any other circuits that need to generate Wake signals from the Suspend-to-RAM state. During S3 (Suspend-to-RAM) all signals attached to powered down planes will be tri-stated or driven low, unless they are pulled using a Pull-up resistor.

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

The SLP\_S4# output signal is used to remove power to additional subsystems that are powered during SLP\_S3#, as well as power to the system memory, since the context of the system is saved on the disk. Cutting power to the memory may be done using the power supply, or by external FETs on the motherboard.

SLP\_S5# output signal can be used to cut power to the system core supply.

### SLP\_S4# and Suspend-to-RAM Sequencing

The system memory suspend voltage regulator is controlled by the Glue logic. The SLP\_S4# signal should be used to remove power to system memory rather than the SLP\_S5# signal. The SLP\_S4# logic in the PCH provides a mechanism to fully cycle the power to the DRAM and/or detect if the power is not cycled for a minimum time.

---

#### NOTE

To use the minimum DRAM power-down feature that is enabled by the SLP\_S4# Assertion Stretch Enable bit (D31:F0:A4h Bit 3), the DRAM power must be controlled by the SLP\_S4# signal.

---

### PCH\_PWROK Signal

When asserted, PCH\_PWROK is an indication to the PCH that its core well power rails are powered and stable. PCH\_PWROK can be driven asynchronously. When PCH\_PWROK is low, the PCH asynchronously asserts PLTRST#. PCH\_PWROK must not glitch, even if RSMRST# is low.

It is required that the power associated with PCIe\* have been valid for 99 ms prior to PCH\_PWROK assertion in order to comply with the 100 ms PCIe\* 2.0 specification on PLTRST# de-assertion.

---

**NOTE**

SYS\_RESET# is recommended for implementing the system reset button. This saves external logic that is needed if the PCH\_PWROK input is used. Additionally, it allows for better handling of the SMBus and processor resets and avoids improperly reporting power failures.

---

**BATLOW# (Battery Low)**

The BATLOW# input can inhibit waking from S3, S4, S5 and Deep Sx states if there is not sufficient power. It also causes an SMI if the system is already in an S0 state.

**SLP\_WLAN# Pin Behavior**

The PCH controls the voltage rails into the external wireless LAN PHY using the SLP\_WLAN# pin.

- The wireless LAN PHY is always powered when the Host is running.
  - SLP\_WLAN#='1' whenever SLP\_S3#='1'.
- If Wake on Wireless LAN (WoWLAN) is required from S3/S4/S5 states, the host BIOS must set HOST\_WLAN\_PP\_EN.
- If WoWLAN is required from Deep Sx, the host BIOS must set DSX\_WLAN\_PP\_EN.
- If Intel® CSE has access to the Wireless LAN device:
  - The Wireless LAN device must always be powered as long as Intel® CSE is powered. SLP\_WLAN#='1' whenever SLP\_A#='1'.
  - If Wake on Wireless LAN (WoWLAN) is required from M-Off state, Intel® CSE will configure SLP\_WLAN#='1' in Sx/M-Off.

Intel® CSE configuration of SLP\_WLAN# in Sx/M-Off is dependent on Intel® CSE power policy configuration.

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

**EXT\_PWR\_GATE# Pin Behavior**

EXT\_PWR\_GATE# can be used to control a FET gating off the HSIO/SRAM power supply to PCH. This provides additional power savings during connected standby states. The ramp time of the FET can be controlled via MODPHY\_PM\_CFG3.

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

**SUSPWRDNACK/SUSWARN#/GPP\_A13 Steady State Pin Behavior**

Below table summarizes SUSPWRDNACK/SUSWARN#/GPP\_A13 pin behavior.

**Table 37. 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 Deep Sx
SUSPWRDNACK	Not Supported	Native	0	Depends on Intel® CSE power package and power source (Note 1)	Off
SUSWARN#	Supported	Native	1	1 (Note 2)	Off
GPP_A13	Do not Care	IN	High-Z	High-Z	Off
	Do not Care	OUT	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® CSE power policy configuration.  
 2. If entering Deep Sx, pin will assert and become undriven ("Off") when suspend well drops upon Deep Sx entry.

**Table 38. 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® CSE power policy configuration.

*Note:* Refer to [Table 39](#) on page 112

**RTCRST# and SRTCST#**

RTCST# 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 RTCST# was asserted and clear internal PCH registers accordingly. It is imperative that this signal not be pulled low in the S0 to S5 states.

SRTCST# is used to reset portions of the Intel® Converged Security 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 RTCST#) 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 RTCST# also being driven low) may cause the platform to enter an indeterminate state. Similar to RTCST#, it is imperative that SRTCST# not be pulled low in the S0 to S5 states.

**10.4.11 Legacy Power Management Theory of Operation**

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

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

### Mobile APM Power Management

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

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

#### 10.4.12 Reset Behavior

When a reset is triggered, the PCH will send a warning message to the processor to allow the processor to attempt to complete any outstanding memory cycles and put memory into a safe state before the platform is reset. When the processor is ready, it will send an acknowledge message to the PCH. Once the message is received the PCH asserts PLTRST#.

The PCH does not require an acknowledge message from the processor to trigger PLTRST#. A global reset will occur after four seconds if an acknowledge from the processor is not received.

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

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

A reset in which the host and Intel® CSE 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® CSE 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# (if pins are not configured as GPIOs), are turned off. All PCH functionality is reset except RTC Power Well backed information and Suspend well status, configuration, and functional logic for controlling and reporting the reset. The host stays there until a valid wake event occurs.

The following table shows the various reset triggers.

**Table 39. Causes of Host and Global Resets**

Trigger	Host Reset Without Power Cycle <sup>1</sup>	Host Reset With Power Cycle <sup>2</sup>	Global Reset With Power Cycle <sup>3</sup>	Straight to S5 <sup>6</sup> (Host Stays There)
Write of 0Eh to CF9h (RST_CNT Register) when CF9h when Global Reset Bit=0b	No	Yes	No <sup>4</sup>	
Write of 06h to CF9h (RST_CNT Register) when CF9h when Global Reset Bit=0b	Yes	No	No <sup>4</sup>	
Write of 06h or 0Eh to CF9h (RST_CNT Register) when CF9h when Global Reset Bit=1b	No	No	Yes	
SYS_RESET# Asserted and CF9h (RST_CNT Register) Bit 3 = 0	Yes	No	No <sup>4</sup>	
SYS_RESET# Asserted and CF9h (RST_CNT Register) Bit 3 = 1	No	Yes	No <sup>4</sup>	
SMBus Target Message received for Reset with Power-Cycle	No	Yes	No <sup>4</sup>	
SMBus Target Message received for Reset without Power-Cycle	Yes	No	No <sup>4</sup>	
SMBus Target Message received for unconditional Power Down	No	No	No	Yes
TCO Watchdog Timer reaches zero two times	Yes	No	No <sup>4</sup>	
Power Failure: PCH_PWROK signal goes inactive in S0 or DSW_PWROK drops	No	No	Yes	
SYS_PWROK Failure: SYS_PWROK signal goes inactive in S0	No	No	Yes	
Processor Thermal Trip (THERMTRIP#) causes transition to S5 and reset asserts	No	No	No	Yes
PCH internal thermal sensors signals a catastrophic temperature condition	No	No	No	Yes
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 <sup>4</sup>	
Special shutdown cycle from processor causes CF9h-like PLTRST# and CF9h Global Reset Bit = 0 and CF9h (RST_CNT Register) Bit 3 = 0	Yes	No	No <sup>4</sup>	
Intel® Converged Security Engine Triggered Host Reset without Power-Cycle	Yes	No	No <sup>4</sup>	
Intel® Converged Security Engine Triggered Host Reset with Power-Cycle	No	Yes	No <sup>4</sup>	
Intel® Converged Security Engine Triggered Power Button Override	No	No	No	Yes

**continued...**



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

*Notes:* 1. The PCH drops this type of reset request if received while the system is in S3/S4/S5.  
 2. PCH does not drop this type of reset request if received while system is in a software-entered S3/S4/S5 state. However, the PCH will perform the reset without executing the RESET\_WARN protocol in these states.  
 3. The PCH does not send warning message to processor, reset occurs without delay.  
 4. Trigger will result in Global Reset with Power-Cycle if the acknowledge message is not received by the PCH.  
 5. The PCH waits for enabled wake event to complete reset.  
 6. Upon entry to S5, if Deep Sx is enabled and conditions are met as per [Deep Sx](#) on page 100, the system will transition to Deep Sx.  
 7. PLTRST# Entry Timeout is automatically initiated if the hardware detects that the PLTRST# sequence has not been completed within 4 seconds of being started.  
 8. Trigger will result in Global Reset with Power-Cycle if AGR\_LS\_EN=1 and Global Reset occurred while the current or destination state was S0.

## 10.5 Advanced Configuration and Power Interface (ACPI) States Supported

This section describes the ACPI states supported by the processor.

**Table 40. System States**

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

*continued...*

State	Description
	Here, SLP_S3, SLP_S4, and SLP_S5 are all active until a wake occurs.
G3	<b>Mechanical OFF:</b> System context not maintained. All power shut except for the RTC. No "Wake" events are possible because the system does not have any power. This state occurs if the user removes the batteries, turns off a mechanical switch, or if the system power supply is at a level that is insufficient to power the "waking" logic. When system power returns the transition will depend on the state just prior to the entry to G3.

**Table 41. Integrated Memory Controller (IMC) States**

State	Description
Power-Up	CKE asserted. Active mode.
Pre-Charge Power Down	CKE de-asserted (not self-refresh) with all banks closed.
Active Power Down	CKE de-asserted (not self-refresh) with minimum one bank active.
Self-Refresh	CKE de-asserted using device self-refresh.

**Table 42. G, S, and C Interface State Combinations**

Global (G) State	Sleep (S) State	Processor Package (C) State	Processor State	System Clocks	Description
G0	S0	C0	Full On	On	Full On
G0	S0	C2 <sup>1</sup>	Deep Sleep	On	Deep Sleep
G0	S0	C3 <sup>1</sup>	Deep Sleep	On	Deep Sleep
G0	S0	C6	Deep Power Down	On	Deep Power Down
G0	S0	C8/C10	Off	On	Deeper Power Down
G1	S3	Power off	Off	Off, except RTC	Suspend to RAM
G1	S4	Power off	Off	Off, except RTC	Suspend to Disk
G2	S5	Power off	Off	Off, except RTC	Soft Off
G3	N/A	Power off	Off	Power off	Hard off

**NOTE**

1. PkgC2/C3 are non-architectural: software cannot request to enter these states explicitly. These states are intermediate states between PkgC0 and PkgC6.

## 10.6 Processor IA Core Power Management

While executing code, Enhanced Intel SpeedStep® Technology and Intel® Speed Shift technology optimizes the processor’s IA core frequency and voltage based on workload. Each frequency and voltage operating point is defined by ACPI as a P-state. When the processor is not executing code, it is idle. A low-power idle state is defined by ACPI as a C-state. In general, deeper power C-states have longer entry and exit latencies.

## 10.6.1 OS/HW Controlled P-states

### 10.6.1.1 Enhanced Intel SpeedStep® Technology

Enhanced Intel SpeedStep® Technology enables OS to control and select P-state. For more information, refer to [Enhanced Intel SpeedStep® Technology](#) on page 63.

### 10.6.1.2 Intel® Speed Shift Technology

Intel® Speed Shift Technology is an energy efficient method of frequency control by the hardware rather than relying on OS control. For more details, refer to [Intel® Speed Shift Technology](#) on page 65.

## 10.6.2 Low-Power Idle States

When the processor is idle, low-power idle states (C-states) are used to save power. More power savings actions are taken for numerically higher C-states. However, deeper C-states have longer exit and entry latencies. Resolution of C-states occurs at the thread, processor IA core, and processor package level.

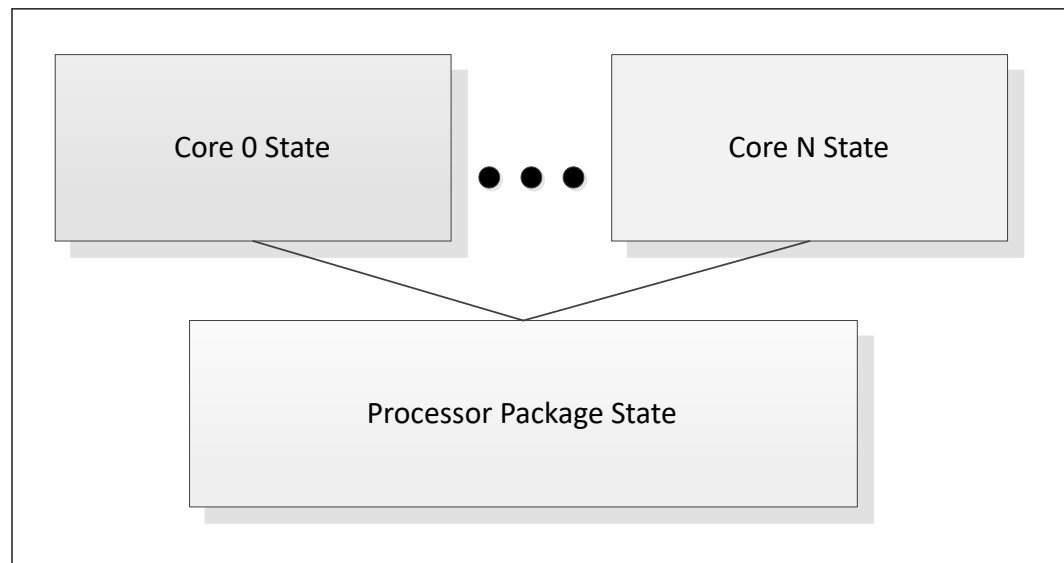
---

### CAUTION

Long-term reliability cannot be assured unless all the Low-Power Idle States are enabled. Refer to the appropriate processor family BIOS Specification for enabling details.

---

**Figure 11. Idle Power Management Breakdown of the Processor IA Cores**



While individual threads can request low-power C-states, power saving actions only take place once the processor IA core C-state is resolved. processor IA core C-states are automatically resolved by the processor. For thread and processor IA core C-states, a transition to and from C0 state is required before entering any other C-state.

### 10.6.3 Requesting the Low-Power Idle States

The primary software interfaces for requesting low-power idle states are through the MWAIT instruction with sub-state hints and the HLT instruction (for C1 and C1E). However, the software may make C-state requests using the legacy method of I/O reads from the ACPI-defined processor clock control registers, referred to as P\_LVLx. This method of requesting C-states provides legacy support for operating systems that initiate C-state transitions using I/O reads.

For legacy operating systems, P\_LVLx I/O reads are converted within the processor to the equivalent MWAIT C-state request. Therefore, P\_LVLx reads do not directly result in I/O reads to the system. The feature, known as I/O MWAIT redirection, should be enabled in the BIOS. To enable it, refer to the appropriate processor family BIOS Specification.

The BIOS can write to the C-state range field of the PMG\_IO\_CAPTURE MSR to restrict the range of I/O addresses that are trapped and emulate MWAIT like functionality. Any P\_LVLx reads outside of this range do not cause an I/O redirection to MWAIT(Cx) like the request. They fall through like a normal I/O instruction.

When P\_LVLx I/O instructions are used, MWAIT sub-states cannot be defined. The MWAIT sub-state is always zero if I/O MWAIT redirection is used. By default, P\_LVLx I/O redirections enable the MWAIT 'break on EFLAGS.IF' feature that triggers a wake up on an interrupt, even if interrupts are masked by EFLAGS.IF.

### 10.6.4 Processor IA Core C-State Rules

The following are general rules for all processor IA core C-states unless specified otherwise:

- A processor IA core C-State is determined by the lowest numerical thread state (such as Thread 0 requests C1E while Thread 1 requests C6 state, resulting in a processor IA core C1E state). Refer to the *G, S, and C Interface State Combinations* table.
- A processor IA core transitions to C0 state when:
  - An interrupt occurs
  - There is an access to the monitored address if the state was entered using an MWAIT/Timed MWAIT instruction
  - The deadline corresponding to the Timed MWAIT instruction expires
- An interrupt directed toward a single thread wakes up only that thread.
- If any thread in a processor IA core is active (in C0 state), the core's C-state will resolve to C0.
- Any interrupt coming into the processor package may wake any processor IA core.
- A system reset re-initializes all processor IA cores.

**Table 43. Core C-states**

Core C-State	C-State Request Instruction	Description
<b>C0</b>	N/A	The normal operating state of a processor IA core where a code is being executed
<b>C1</b>	MWAIT(C1)	AutoHalt - core execution stopped, autonomous clock gating (package in C0 state)
<b>C1E</b>	MWAIT(C1E)	Core C1 + lowest frequency and voltage operating point (package in C0 state)
<b>C6-C10</b>	MWAIT(C6/C8/10) or IO read=P_LVL3//6/8	Processor IA, flush their L1 instruction cache, the L1 data cache, and L2 cache to the LLC shared cache cores save their architectural state to an SRAM before reducing IA cores voltage, if possible may also be reduced to 0V. Core clocks are off.

### Core C-State Auto-Demotion

In general, deeper C-states, such as C6, have long latencies and have higher energy entry/exit costs. The resulting performance and energy penalties become significant when the entry/exit frequency of a deeper C-state is high. Therefore, incorrect or inefficient usage of deeper C-states have a negative impact on battery life and idle power. To increase residency and improve battery life and idle power in deeper C-states, the processor supports C-state auto-demotion.

C-State auto-demotion:

- C6 to C1/C1E

The decision to demote a processor IA core from C6 to C1/C1E is based on each processor IA core's immediate residency history. Upon each processor IA core C6 request, the processor IA core C-state is demoted to C1 until a sufficient amount of residency has been established. At that point, a processor IA core is allowed to go into C6. If the interrupt rate experienced on a processor IA core is high and the processor IA core is rarely in a deep C-state between such interrupts, the processor IA core can be demoted to a C1 state.

This feature is disabled by default. BIOS should enable it in the PMG\_CST\_CONFIG\_CONTROL register. The auto-demotion policy is also configured by this register. Refer to the appropriate processor family BIOS Specification for more details.

### 10.6.5 Package C-States

The processor supports C0, C2, C3, C6, C8, and C10 package states. The following is a summary of the general rules for package C-state entry. These apply to all package C-states, unless specified otherwise:

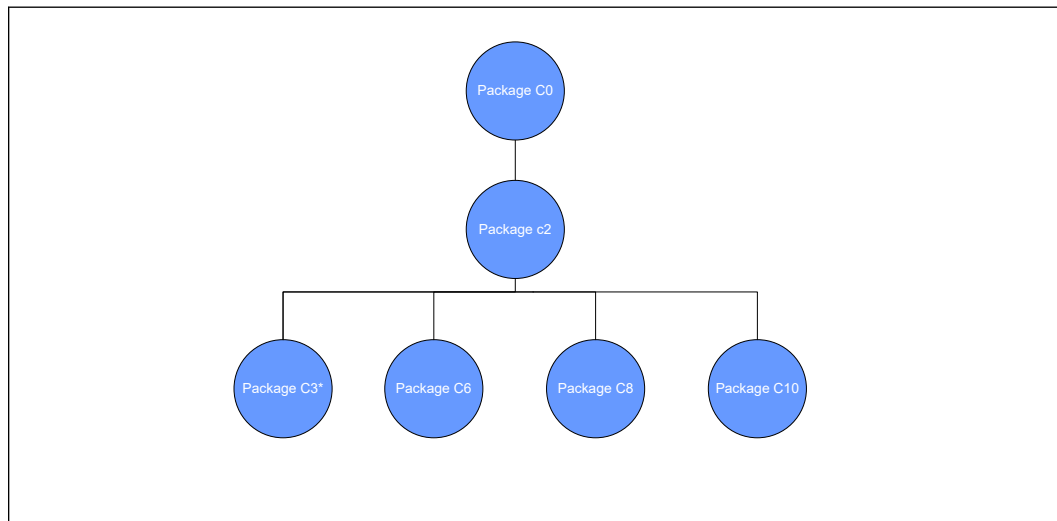
- A package C-state request is determined by the lowest numerical processor IA core C-state amongst all processor IA cores.
- A package C-state is automatically resolved by the processor depending on the processor IA core idle power states and the status of the platform components.
  - Each processor IA core can be at a lower idle power state than the package if the platform does not grant the processor permission to enter a requested package C-state.

- The platform may allow additional power savings to be realized in the processor.
- For package C-states, the processor is not required to enter C0 before entering any other C-state.
- Entry into a package C-state may be subject to auto-demotion – that is, the processor may keep the package in a deeper package C-state then requested by the operating system if the processor determines, using heuristics, that the deeper C-state results in better power/performance.

The processor exits a package C-state when a break event is detected. Depending on the type of break event, the processor does the following:

- If a processor IA core break event is received, the target processor IA core is activated and the break event message is forwarded to the target processor IA core.
  - If the break event is not masked, the target processor IA core enters the processor IA core C0 state and the processor enters package C0.
  - If the break event is masked, the processor attempts to re-enter its previous package state.
- If the break event was due to a memory access or snoop request,
  - But the platform did not request to keep the processor in a higher package C-state, the package returns to its previous C-state.
  - And the platform requests a higher power C-state, the memory access or snoop request is serviced and the package remains in the higher power C-state.

**Figure 12. Package C-State Entry and Exit**



PKG C2 and C3 can not be requested explicitly by the software

**Table 44. Package C-States**

Package C state	Description	Dependencies
<b>PKG C0</b>	Processor active state. At least one IA core in C0. Processor Graphic in RC0 (Graphics active state) or RC6 (Graphics Core power down state).	-
<b>PKG C2</b>	Cannot be requested explicitly by the Software. All processor IA cores in C6 or deeper + Processor Graphic cores in RC6, memory path may be open. The processor will enter Package C2 when: <ul style="list-style-type: none"> <li>Transitioning from Package C0 to deep Package C state or from deep Package C state to Package C0.</li> <li>All IA cores requested C6 or deeper + Processor Graphic cores in RC6 but there are constraints (LTR, programmed timer events in the near future and so forth) prevent entry to any state deeper than C2 state.</li> <li>All IA cores requested C6 or deeper + Processor Graphic cores in RC6 but a device memory access request is received. Upon completion of all outstanding memory requests, the processor transitions back into a deeper package C-state.</li> </ul>	All processor IA cores in C6 or deeper. Processor Graphic cores in RC6.
<b>PKG C3</b>	Cannot be requested explicitly by the Software. All cores in C6 or deeper + Processor Graphics in RC6, LLC may be flushed and turned off, memory in self refresh, memory clock stopped. The processor will enter Package C3 when: <ul style="list-style-type: none"> <li>All IA cores in C6 or deeper + Processor Graphic cores in RC6.</li> <li>The platform components/devices allows proper LTR for entering Package C3.</li> </ul>	All processor IA cores in C6 or deeper. Processor Graphics in RC6. memory in self refresh, memory clock stopped. LLC may be flushed and turned off.
<b>PKG C6</b>	Package C3 + BCLK is off + IMVP VRs voltage reduction/PSx state is possible. The processor will enter Package C6 when: <ul style="list-style-type: none"> <li>All IA cores in C6 or deeper + Processor Graphic cores in RC6.</li> <li>The platform components/devices allow proper LTR for entering Package C6.</li> </ul>	Package C3. BCLK is off. IMVP VRs voltage reduction/PSx state is possible.
<b>PKG C8</b>	Of all IA cores requested C8 + LLC should be flushed at once, voltage will be removed from the LLC. The processor will enter Package C8 when: <ul style="list-style-type: none"> <li>All IA cores in C8 or deeper + Processor Graphic cores in RC6.</li> <li>The platform components/devices allow proper LTR for entering Package C8.</li> </ul>	Package C6 If all IA cores requested C8, LLC is flushed in a single step, voltage will be removed from the LLC.
<b>PKG C10</b>	Package C8 + display in PSR or powered, ff all VRs at PS4 or LPM + crystal clock off. The processor will enter Package C10 when: <ul style="list-style-type: none"> <li>All IA cores in C10 + Processor Graphic cores in RC6.</li> <li>The platform components/devices allow proper LTR for entering Package C10.</li> </ul>	Package C8. All IA cores in C8 or deeper. Display in PSR or powered off <sup>1</sup> . All VRs at PS4 or LPM. Crystal clock off.

*Note:* Display In PSR is only on single embedded panel configuration and panel support PSR feature.

**Package C-State Auto-Demotion**

The Processor may demote the Package C state to a shallower C state, for example instead of going into package C10, it will demote to package C8 (and so on as required). The processor decision to demote the package C state is based on the required C states latencies, entry/exit energy/power and devices LTR.

**Modern Standby**

Modern Standby is a platform state. On display time out the OS requests the processor to enter package C10 and platform devices at RTD3 (or disabled) in order to attain low power in idle. Modern Standby requires proper BIOS (refer to BIOS specification ) and OS configuration.

**Dynamic LLC Sizing**

When all processor IA cores request C8 or deeper C-state, internal heuristics dynamically flushes the LLC. Once the processor IA cores enter a deep C-state, depending on their MWAIT sub-state request, the LLC is either gradually flushed N-ways at a time or flushed all at once. Upon the processor IA cores exiting to C0 state, the LLC is gradually expanded based on internal heuristics.

**10.6.6 Package C-States and Display Resolutions**

The integrated graphics engine has the frame buffer located in system memory. When the display is updated, the graphics engine fetches display data from system memory. Different screen resolutions and refresh rates have different memory latency requirements. These requirements may limit the deepest Package C-state the processor can enter. Other elements that may affect the deepest Package C-state available are the following:

- Display is on or off
- Single or multiple displays
- Native or non-native resolution
- Panel Self Refresh (PSR) technology

**NOTE**

Display resolution is not the only factor influencing the deepest Package C-state the processor can get into. Device latencies, interrupt response latencies, and core C-states are among other factors that influence the final package C-state the processor can enter.

The following table lists display resolutions and deepest available package C-State. The display resolutions are examples using common values for blanking and pixel rate. Actual results will vary. The table shows the deepest possible Package C-state. System workload, system idle, and AC or DC power also affect the deepest possible Package C-state.

**Table 45. Deepest Package C-State Available**

Resolution	Number of Displays
Up to 4096x2304 60Hz	Single
<i>Notes:</i> 1. All Deep states are with Display ON. 2. The deepest C-state has variance, dependent various parameters such SW and Platform devices.	



## 10.7 Processor Graphics Power Management

### 10.7.1 Memory Power Savings Technologies

#### Intel® Rapid Memory Power Management (Intel® RMPM)

Intel® Rapid Memory Power Management (Intel® RMPM) conditionally places memory into self-refresh when the processor is in package C3 or deeper power state to allow the system to remain in the deeper power states longer for memory not reserved for graphics memory. Intel® RMPM functionality depends on graphics/display state (relevant only when processor graphics is being used), as well as memory traffic patterns generated by other connected I/O devices.

### 10.7.2 Display Power Savings Technologies

#### Intel® Seamless Display Refresh Rate Switching Technology (Intel® SDRRS Technology) with eDP\* Port

Intel® DRRS provides a mechanism where the monitor is placed in a slower refresh rate (the rate at which the display is updated). The system is smart enough to know that the user is not displaying either 3D or media like a movie where specific refresh rates are required. The technology is very useful in an environment such as a plane where the user is in battery mode doing E-mail, or other standard office applications. It is also useful where the user may be viewing web pages or social media sites while in battery mode.

#### Intel® Automatic Display Brightness

Intel® Automatic Display Brightness feature dynamically adjusts the back-light brightness based upon the current ambient light environment. This feature requires an additional sensor to be on the panel front. The sensor receives the changing ambient light conditions and sends the interrupts to the Intel Graphics driver. As per the change in Lux, (current ambient light luminance), the new back-light setting can be adjusted through BLC (Back Light Control). The converse applies for a brightly lit environment. Intel® Automatic Display Brightness increases the back-light setting.

#### Smooth Brightness

The Smooth Brightness feature is the ability to make fine grained changes to the screen brightness. All Windows\* 11 system that support brightness control are required to support Smooth Brightness control and it should be supporting 101 levels of brightness control. Apart from the Graphics driver changes, there may be few System BIOS changes required to make this feature functional.

#### Intel® Display Power Saving Technology (Intel® DPST) 7.0

The Intel® DPST technique achieves back-light power savings while maintaining a good visual experience. This is accomplished by adaptively enhancing the displayed image while decreasing the back-light brightness simultaneously. The goal of this technique is to provide equivalent end-user-perceived image quality at a decreased back-light power level.

1. The original (input) image produced by the operating system or application is analyzed by the Intel® DPST subsystem. An interrupt to Intel® DPST software is generated whenever a meaningful change in the image attributes is detected. (A

meaningful change is when the Intel® DPST software algorithm determines that enough brightness, contrast, or color change has occurred to the displaying images that the image enhancement and back-light control needs to be altered.)

2. Intel® DPST subsystem applies an image-specific enhancement to increase image contrast, brightness, and other attributes.
3. A corresponding decrease to the back-light brightness is applied simultaneously to produce an image with similar user-perceived quality (such as brightness) as the original image.

Intel® DPST 6.3 has improved power savings without adversely affecting the performance.

### Panel Self-Refresh 2 (PSR 2)

Panel Self-Refresh feature allows the Processor Graphics core to enter low-power state when the frame buffer content is not changing constantly. This feature is available on panels capable of supporting Panel Self-Refresh. Apart from being able to support, the eDP\* panel should be eDP 1.4 compliant. PSR 2 adds partial frame updates and requires an eDP 1.4 compliant panel.

### Low-Power Single Pipe (LPSP)

Low-power single pipe is a power conservation feature that helps save power by keeping the inactive pipes powered OFF. This feature is enabled only in a single display configuration without any scaling functionalities. LPSP is achieved by keeping a single pipe enabled during eDP\* only with minimal display pipeline support. This feature is panel independent and works with any eDP panel (port A) in single display mode.

### Intel® Smart 2D Display Technology (Intel® S2DDT)

Intel® S2DDT reduces display refresh memory traffic by reducing memory reads required for display refresh. Power consumption is reduced by less accesses to the IMC. Intel S2DDT is only enabled in single pipe mode.

Intel® S2DDT is most effective with:

- Display images well suited to compression, such as text windows, slide shows, and so on. Poor examples are 3D games.
- Static screens such as screens with significant portions of the background showing 2D applications, processor benchmarks, and so on, or conditions when the processor is idle. Poor examples are full-screen 3D games and benchmarks that flip the display image at or near display refresh rates.

## 10.7.3 Processor Graphics Core Power Savings Technologies

### Intel® Graphics Dynamic Frequency

Intel® Turbo Boost Technology 2.0 is the ability of the processor IA cores and graphics (Graphics Dynamic Frequency) cores to opportunistically increase frequency and/or voltage above the guaranteed processor and graphics frequency for the given part. Intel® Graphics Dynamic Frequency is a performance feature that makes use of unused package power and thermals to increase application performance. The increase in frequency is determined by how much power and thermal budget is available in the package, and the application demand for additional processor or graphics performance. The processor IA core control is maintained by an embedded controller. The graphics driver dynamically adjusts between P-States to maintain

optimal performance, power, and thermals. The graphics driver will always place the graphics engine in its lowest possible P-State. Intel® Graphics Dynamic Frequency requires BIOS support. Additional power and thermal budget should be available.

**Intel® Graphics Render Standby Technology (Intel® GRST)**

Intel® Graphics Render Standby Technology is a technique designed to optimize the average power of the graphics part. The Graphics Render engine will be put in a sleep state, or Render Standby (RS), during times of inactivity or basic video modes. While in Render Standby state, the graphics part will place the VR (Voltage Regulator) into a low voltage state. Hardware will save the render context to the allocated context buffer when entering RS state and restore the render context upon exiting RS state.

**Dynamic FPS (DFPS)**

Dynamic FPS (DFPS) or dynamic frame-rate control is a runtime feature for improving power-efficiency for 3D workloads. Its purpose is to limit the frame-rate of full screen 3D applications without compromising on user experience. By limiting the frame rate, the load on the graphics engine is reduced, giving an opportunity to run the Processor Graphics at lower speeds, resulting in power savings. This feature works in both AC/DC modes.

**10.8 System Agent Enhanced Intel SpeedStep® Technology**

System Agent Enhanced Intel SpeedStep® Technology is a dynamic voltage frequency scaling of the System Agent clock based on memory utilization. Unlike processor core and package Enhanced Intel SpeedStep® Technology, System Agent Enhanced Intel SpeedStep® Technology has three valid operating points. When running light workload and SA Enhanced Intel SpeedStep® Technology is enabled, the DDR data rate may change as follows:

Before changing the DDR data rate, the processor sets DDR to self-refresh and changes the needed parameters. The DDR voltage remains stable and unchanged.

BIOS/MRC DDR training at maximum, mid and minimum frequencies sets I/O and timing parameters.

Refer to BIOS Specification for more information.

**10.9 Type C Sub System (TCSS) Power State**

**Table 46. TCSS Power State**

TCSS Power State	Allowed Package C Status	Device Attached	Description
TC0	PC0-PC3	Yes	xHCI, xDCI controllers may be active. DMA may be active.
TC7	PC6-PC10	Yes	xHCI and xDCI are in D3. Controller is in D3 or D0 idle.
TC-Cold	PC3-PC10	No	xHCI / xDCI are in D3 IOM is active
TC10	PC6-PC10	No	Deepest Power state

*continued...*

TCSS Power State	Allowed Package C Status	Device Attached	Description
			xHCI and xDCI are in D3. IOM is inactive
<p><i>Notes:</i></p> <ul style="list-style-type: none"> <li>• IOM - TCSS Input Output Manager:               <ul style="list-style-type: none"> <li>— The IOM interacts with the processor to perform power management, boot, reset, connect and disconnect devices to TYPE-C sub-system</li> </ul> </li> <li>• TCSS Devices (xHCI / xDCI) - power states:               <ul style="list-style-type: none"> <li>— D0 - Device at Active state.</li> <li>— D3 - Device at lowest-powered state.</li> </ul> </li> </ul>			

## 11.0 Power Delivery

This section provides information on the following topics:

- Power and Ground Signals
- FIVR

### 11.1 Power and Ground Signals

This section describes the power rails.

**Table 47. Power Rail Descriptions**

Name	Description
VCC_VNNEXT_1P05	Used for FIVR PRIM_CORE bypass mode during S0ix and Sx: 1.05 V
VCC_V1P05EXT_1P05	Used for FIVR PCH IO bypass mode during S0ix and Sx: 1.05 V
VCCA_CLKLDO_1P8	Analog supply for internal clocks: 1.8 V
VCCPRIM1P05_OUT_PCH	1.05 V Primary Well: for CNVi and other internal I/O blocks.
VCCDSW_1P05	Deep Sx Well: 1.05 V. This rail is generated by on die DSW low dropout (LDO) linear regulator to supply DSW core logic.
VCCPRIM_1P8	1.8 V Primary Well.
VCCPRIM_3P3	3.3 V Primary Well.
VCCPGPPR	Audio Power 3.3 V or 1.8 V. If powered at 3.3 V, the 3.3 V supply can come from VCCPRIM_3P3 supply. If powered at 1.8 V, the 1.8 V supply can come from VCCPRIM_1P8 supply.
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. <i>Notes:</i> 1. VCCRTC nominal voltage is 3.0 V. This rail is intended to always come up first and always stay on. It should NOT be power cycled regularly on non-coin battery designs. 2. Implementation should not attempt to clear CMOS by using a jumper to pull VCCRTC low. Clearing CMOS can be done by using a jumper on RTCRST# or GPI.
VCCDPHY_1P24	1.24 V for CNVi logic. This rail is generated internally with a LDO and needs to be routed to the motherboard so that the rail can be supplied back to the processor.
VCCLDOSTD_0P85	This rail is generated internally and needs to be routed out to the motherboard for decoupling purpose.
VCC1P05_OUT_FET	FIVR output rail: 1.05 V, used for CPU rails VCCST/STG.
VSS	Ground
VCCCORE	Processor IA Cores and Ring power rail
VCCGT	Processor Graphics power rail
VCCANA	Support internal Analog Rails, TCSS, Display, PCIE and other internal Blocks.
VCCIN_AUX	Support internal FIVR's, SA, PCIE, Display IO and other internal Blocks.

*continued...*

Name	Description
VCCIN_AUX_FLTR	Support internal FIVR's, SA, PCIe, Display IO and other internal Blocks. this pin should be connected to decoupling for filter.
VCC1P05_PROC	Sustain and Sustain Gated Power Rail
VDD2	System Memory power rail
VCCGT_SENSE	Isolated, low impedance voltage sense pins. They can be used to sense or measure voltage near the silicon.
VCC_SENSE	
VCCANA_SENSE	
VCCINAUX_SENSE	
VCC_MIPILP	<i>Note:</i> When MIPI DSI interface is been used, <b>VCC_MIPILP</b> should be connected to 1.26V (on board VR).

**Table 48. Processor Ground Rails Signals**

Signal Name	Description	Dir.	Buffer Type	Link Type
VSSGT_SENSE	Isolated, low impedance Ground sense pins. They can be used for the reference ground near the silicon.	N/A	GND_SENSE	—
VSS_SENSE				
VSSANA_SENSE				
VSSINAUX_SENSE				

## 11.2 FIVR

PCH integrates multiple voltage rails onto the PCH in order to reduce BOM costs for the platform and to enable additional voltage level features.

There are FIVRs integrated on the PCH, VNN/V1P05 which is sourced from VCCIN\_Aux.

## 11.3 PCH Platform Voltage Rails

**Table 49. PCH Platform Voltage Rails**

Power Rail	Voltage	Description
VCCIN_AUX	1.65 V or 1.8 V - Active 1.10 V - Retention OFF - Idle States	PCH FIVR Input rail
VCCPRIM_1P8	1.8 V	Primary well supply
VCCDSW_3P3	3.3 V	Deep sleep well supply, 3.3 V
VCCPRIM_3P3	3.3 V	Primary well supply, 3.3 V
<i>continued...</i>		

Power Rail	Voltage	Description
VCCRTC	3.3 V	RTC supply
VCC_V1P05EXT_1P05 (Optional)	1.05 V	Used during Sx and S0ix modes for bypassing the FIVR internal supply
VCC_VNNEXT_1P05 (Optional)	0.7 V 0.78 V 1.05 V (not used for S0ix)	Used during Sx and S0ix modes for bypassing the FIVR internal supply

### VCCIN\_AUX

VCCIN\_AUX is the input rail to FIVR. During the deep S0ix states and Sx states, the input rail to the FIVRs can be disabled. This will be done by driving the CORE\_VID values to '00.

VCCIN\_AUX powergood during initial reset is tied into the RSMRST# signal, requiring that the FIVR input voltage rail is stable in the same window as the other SLP\_SUS# rails.

To support dynamic switching during run time of the input VR, CORE\_VID[1:0] pins are driven out from PCH.

### VCCIN\_AUX Control - CORE\_VID Pins

The CORE\_VID pins are used to control the VCCIN\_AUX rail.

**Table 50. CORE\_VID Signaling**

SLP_SUS#	CORE_VID1	CORE_VID0	SLP_S0#	CPU Requirement	VCCIN_AUX Voltage	Comments
0	X	X	X	OFF	OFF	FIVR Input is OFF
1	0	0	0	VCCIN_AUX = 0	0 V	Typically used during S0ix states.
1	0	1	1	VCCIN_AUX = 0	1.10 V	Retention FIVR voltage, no VCCIN_AUX FIVRs active in CPU.
1	1	0	1	VCCIN_AUX = 1.65 V	1.65 V	Low Current Mode Voltage 1.65 V
1	1	1	1	VCCIN_AUX = 1.8 V	1.8 V	High Current Mode Voltage 1.8 V

The default value for CORE\_VID1/0 is 2'b11 (signaling 1.8 V). VCCIN\_AUX configurations are specified through VCCIN\_AUX\_CFG1 & CFG2 registers. In a resume from 0 V, the field in VCCIN\_AUX\_CFG2 will specify the time to resume to 1.8 V.

### NOTE

Leakage from VCCIN\_AUX is expected behavior when CORE\_VID[1:0]=00; this leakage voltage may be as high as 1.15 V during Sx and S0ix states.

### External Bypass Rails Control

Both VCC\_VNNEXT\_1P05 and VCC\_V1P05EXT\_1P05 rails have the ability to have an external bypass to be used when the platform is in S0ix and Sx states. These rails are always on and must be come up after the 1.8 V rail has been brought up.

The external bypass rails can be controlled via below pins without requiring BIOS to be involved during the S0ix states. The pin is VNN\_CTRL - Control of the VCC\_VNNEXT\_1P05 voltage

**Table 51. VNN\_CTRL Pin States**

Platform State	VNN_CTRL	VCC_VNNEXT_1P05
S0	0	0.78 V
S0i2.x	0	0.78 V
S0i3.x	1	0.7 V
Sx	0	1.05 V

#### NOTE

Leakage from VCC\_VNNEXT\_1P05 and VCC\_V1P05EXT\_1P05 power rails may back drive the external bypass voltage regulators (VR) when they are not in use, and VRs output may float up as high as 1.125 V. This is an expected behavior. Intel recommends to select the bypass VRs with an Over Voltage Protection (OVP) threshold that is above 1.125 V for all VCC\_VNNEXT\_1P05 and VCC\_V1P05EXT\_1P05 voltage settings to avoid false VRs shutdown.



## 12.0 Thermal Management

---

### 12.1 Processor Thermal Management

The thermal solution provides both component-level and system-level thermal management. To allow optimal operation and long-term reliability of Intel processor-based systems, the system/processor thermal solution should be designed so that the processor:

- Remains below the maximum junction temperature ( $T_{jMAX}$ ) specification at the maximum Processor Base Power (a.k.a TDP).
- Conforms to system constraints, such as system acoustics, system skin-temperatures, and exhaust-temperature requirements.

---

#### CAUTION

Thermal specifications given in this chapter are on the component and package level and apply specifically to the processor. Operating the processor outside the specified limits may result in permanent damage to the processor and potentially other components in the system.

---

#### 12.1.1 Thermal Considerations

The Processor Base Power (a.k.a TDP) is the maximum sustained power that should be used as a baseline value for the processor thermal solution. Processor Base Power (a.k.a TDP) is the average power dissipation and junction temperature operating condition limit, specified in this document, that is validated during manufacturing for the base configuration when executing a near worst case commercially available workload as specified by Intel for the SKU segment. Processor Base Power (a.k.a TDP) may be exceeded for short periods of time or if running a very high power workload.

The processor integrates multiple processing IA cores, graphics cores and a PCH on a single package. This may result in power distribution differences across the package and should be considered when designing the thermal solution.

Intel® Turbo Boost Technology 2.0 allows processor IA cores to run faster than the base frequency. It is invoked opportunistically and automatically as long as the processor is conforming to its temperature, power delivery, and current control limits. When Intel® Turbo Boost Technology 2.0 is enabled:

- Applications are expected to run closer to Processor Base Power (a.k.a TDP) more often as the processor will attempt to maximize performance by taking advantage of estimated available energy budget in the processor package.
- The processor may exceed the Processor Base Power (a.k.a TDP) for short durations to utilize any available thermal capacitance within the thermal solution. The duration and time of such operation can be limited by platform runtime configurable registers within the processor.

- Graphics peak frequency operation is based on the assumption of only one of the graphics domains (GT/GTx) being active. This definition is similar to the IA core Turbo concept, where peak turbo frequency can be achieved when only one IA core is active. Depending on the workload being applied and the distribution across the graphics domains the user may not observe peak graphics frequency for a given workload or benchmark.
- Thermal solutions and platform cooling that are designed to less than thermal design guidance may experience thermal and performance issues.

---

**NOTE**

Intel® Turbo Boost Technology 2.0 availability may vary between the different SKUs.

---

### 12.1.1.1 Package Power Control

The package power control settings of PL1, PL2, PL3, PL4, and Tau allow the designer to configure Intel® Turbo Boost Technology 2.0 to match the platform power delivery and package thermal solution limitations.

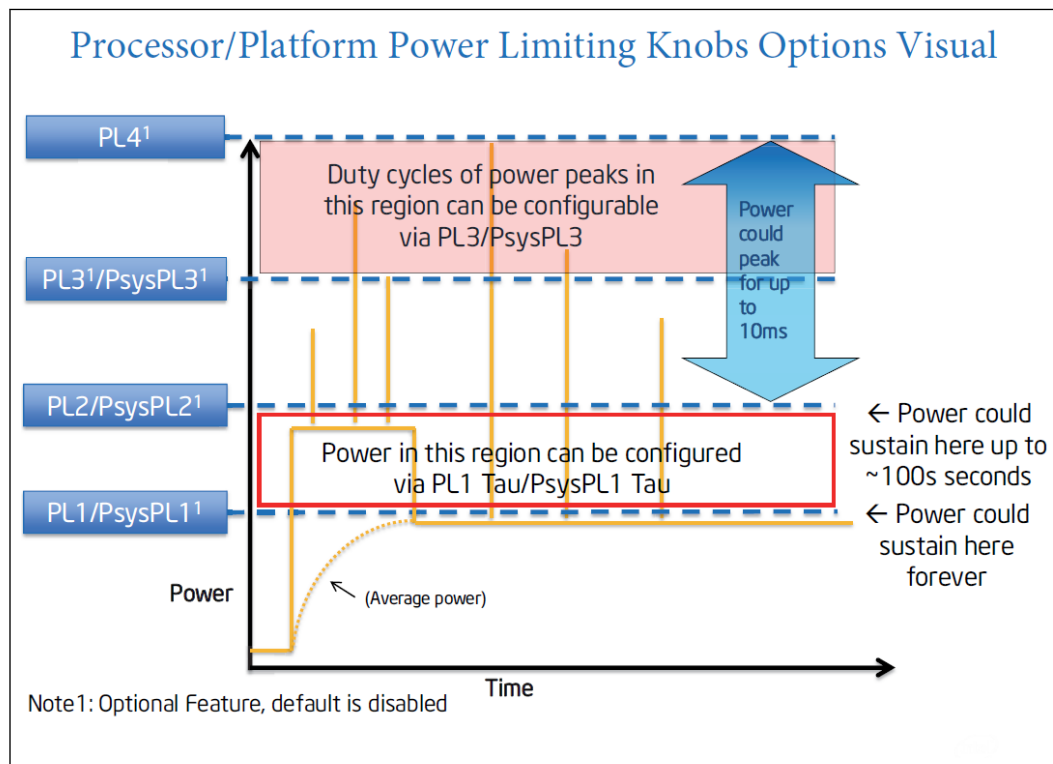
- **Power Limit 1 (PL1):** A threshold for average power that will not exceed - recommend to set to equal TDP power. PL1 should not be set higher than thermal solution cooling limits.
- **Power Limit 2 (PL2):** A threshold that if exceeded, the PL2 rapid power limiting algorithms will attempt to limit the spike above PL2. PL2 should not be set higher than maximum, non-transient capabilities of the processor's power supplies.
- **Power Limit 3 (PL3):** A threshold that if exceeded, the PL3 rapid power limiting algorithms will attempt to limit the duty cycle of spikes above PL3 by reactively limiting frequency. This is an optional setting for battery powered systems to reduce stress on the main battery.
- **Power Limit 4 (PL4):** A limit that will not be exceeded, the PL4 power limiting algorithms will preemptively limit frequency to prevent spikes above PL4. This is an optional setting for battery powered systems to reduce stress on the DC power supply.
- **Turbo Time Parameter (Tau):** An averaging constant used for PL1 Exponential Weighted Moving Average (EWMA) power calculation.

---

**NOTES**

1. Implementation of Intel® Turbo Boost Technology 2.0 only requires configuring PL1 and PL2.
  2. PL3 and PL4 are disabled by default.
-

Figure 13. Package Power Control



### 12.1.1.2 Platform Power Control

The processor supports Psys (Platform Power) to enhance processor power management. The Psys signal needs to be sourced from a compatible charger circuit and routed to the IMVP9.1 (voltage regulator). This signal will provide the total thermally relevant platform power consumption (processor and rest of platform) via SVID to the processor.

When the Psys signal is properly implemented, the system designer can utilize the package power control settings of PsysPL1/Tau, PsysPL2, and PsysPL3 for additional manageability to match the platform power delivery and platform thermal solution limitations for Intel® Turbo Boost Technology 2.0. The operation of the PsysPL1/Tau, PsysPL2 and PsysPL3 are analogous to the processor power limits described in [Package Power Control](#) on page 130.

- **Platform Power Limit 1 (PsysPL1):** A threshold for average platform power that will not be exceeded - recommend to set to equal platform thermal capability.
- **Platform Power Limit 2 (PsysPL2):** A threshold that if exceeded, the PsysPL2 rapid power limiting algorithms will attempt to limit the spikes above PsysPL2.
- **Platform Power Limit 3 (PsysPL3):** A threshold that if exceeded, the PsysPL3 rapid power limiting algorithms will attempt to limit the duty cycle of spikes above PsysPL3 by reactively limiting frequency.
- **PsysPL1 Tau:** An averaging constant used for PsysPL1 Exponential Weighted Moving Average (EWMA) power calculation.

- The Psys signal and associated power limits / Tau are optional for the system designer and disabled by default.
- The Psys data will not include power consumption for charging.
- The Intel Dynamic Tuning Technology (DTT/DPTF) is recommended for performance improvement in mobile platforms. Dynamic Tuning is configured by system manufacturers dynamically optimizing the processor power based on the current platform thermal and power delivery conditions. Contact Intel Representatives for enabling details.

### 12.1.1.3 Turbo Time Parameter (Tau)

Turbo Time Parameter (Tau) is a mathematical parameter (units of seconds) that controls the Intel® Turbo Boost Technology 2.0 algorithm. During a maximum power turbo event, the processor could sustain PL2 for a duration longer than the Turbo Time Parameter. If the power value and/or Turbo Time Parameter is changed during runtime, it may take some time based on the new Turbo Time Parameter level for the algorithm to settle at the new control limits. The time varies depending on the magnitude of the change, power limits and other factors. There is an individual Turbo Time Parameter associated with Package Power Control and Platform Power Control.

### 12.1.2 Assured Power (cTDP) and Low Power Mode (LPM)

Assured Power (cTDP) and Low-Power Mode (LPM) form a design option where the processor's behavior and package Processor Base Power (a.k.a TDP) are dynamically adjusted to a desired system performance and power envelope. Assured Power (cTDP) and Low-Power Mode technologies offer opportunities to differentiate system design while running active workloads on select processor SKUs through scalability, configuration and adaptability. The scenarios or methods by which each technology is used are customizable but typically involve changes to PL1 and associated frequencies for the scenario with a resultant change in performance depending on system's usage. Either technology can be triggered by (but are not limited to) changes in OS power policies or hardware events such as docking a system, flipping a switch or pressing a button. cTDP and LPM are designed to be configured dynamically and do not require an operating system reboot.

---

**NOTE**

Assured Power (cTDP) and Low-Power Mode technologies are not battery life improvement technologies.

---

#### 12.1.2.1 Assured Power (cTDP)

---

**NOTE**

Assured Power (cTDP) availability may vary between the different SKUs.

---

With Assured Power (cTDP), the processor is now capable of altering the maximum sustained power with an alternate processor IA core base frequency. Assured Power (cTDP) allows operation in situations where extra cooling is available or situations where a cooler and quieter mode of operation is desired. Refer to the appropriate processor family BIOS Specification for more enabling details.

cTDP consists of three modes as shown in the following table.

**Table 52. Assured Power (cTDP) Modes**

Mode	Description
Processor Base Power	The average power dissipation and junction temperature operating condition limit for which the processor is validated during manufacturing when executing an associated Intel-specified high-complexity workload at the processor IA core frequency corresponding to the configuration and SKU.
Maximum Assured Power	The SKU-specific processor IA core frequency where manufacturing confirms logical functionality within the set of operating condition limits specified for the SKU segment and Configurable Maximum Assured Power configuration. The Maximum Assured Power Frequency and corresponding Processor Base Power (a.k.a TDP) is higher than the processor IA core Base Frequency and SKU Segment Processor Base Power (a.k.a TDP).
Minimum Assured Power	The processor IA core frequency where manufacturing confirms logical functionality within the set of operating condition limits specified for the SKU segment and Configurable Minimum Assured Power configuration. The Minimum Assured Power Frequency and corresponding Processor Base Power (a.k.a TDP) is lower than the processor IA core Base Frequency and SKU Segment Processor Base Power (a.k.a TDP).

In each mode, the Intel® Turbo Boost Technology 2.0 power limits are reprogrammed along with a new OS controlled frequency range. The Intel Dynamic Tuning Technology driver assists in Processor Base Power (a.k.a TDP) operation by adjusting processor PL1 dynamically. The Assured Power (cTDP) mode does not change the maximum per-processor IA core turbo frequency.

**12.1.2.2 Low Power Mode**

Low-Power Mode (LPM) can provide cooler and quieter system operation. By combining several active power limiting techniques, the processor can consume less power while running at equivalent low frequencies. Active power is defined as processor power consumed while a workload is running and does not refer to the power consumed during idle modes of operation. LPM is only available using the Intel® Dynamic Tuning Technology (Intel® DTT/Intel® DPTF) driver.

Through the Intel® Dynamic Tuning Technology (Intel® DTT/Intel® DPTF) driver, LPM can be configured to use each of the following methods to reduce active power:

- Restricting package power control limits and Intel® Turbo Boost Technology availability
- Off-Lining processor IA core activity (Move processor traffic to a subset of cores)
- Placing a processor IA Core at LFM or LSF (Lowest Supported Frequency)
- Utilizing IA clock modulation
- LPM power as listed in the *TDP Specifications* table is defined at a point which processor IA core working at LSF, GT = RPN and 1 IA core active

Off-lining processor IA core activity is the ability to dynamically scale a workload to a limited subset of cores in conjunction with a lower turbo power limit. It is one of the main vectors available to reduce active power. However, not all processor activity is ensured to be able to shift to a subset of cores. Shifting a workload to a limited subset of cores allows other processor IA cores to remain idle and save power. Therefore, when LPM is enabled, less power is consumed at equivalent frequencies.

Minimum Frequency Mode (MFM) of operation, which is the Lowest Supported Frequency (LSF) at the LFM voltage, has been made available for use under LPM for further reduction in active power beyond LFM capability to enable cooler and quieter modes of operation.

### 12.1.3 Thermal Management Features

Occasionally the processor may operate in conditions that are near to its maximum operating temperature. This can be due to internal overheating or overheating within the platform. In order to protect the processor and the platform from thermal failure, several thermal management features exist to reduce package power consumption and thereby temperature in order to remain within normal operating limits. Furthermore, the processor supports several methods to reduce memory power.

#### 12.1.3.1 Adaptive Thermal Monitor

The purpose of the Adaptive Thermal Monitor is to reduce processor IA core power consumption and temperature until it operates below its maximum operating temperature. Processor IA core power reduction is achieved by:

- Adjusting the operating frequency (using the processor IA core ratio multiplier) and voltage.
- Modulating (starting and stopping) the internal processor IA core clocks (duty cycle).

The Adaptive Thermal Monitor can be activated when the package temperature, monitored by any Digital Thermal Sensor (DTS), meets its maximum operating temperature. The maximum operating temperature implies maximum junction temperature  $T_{jMAX}$ .

Reaching the maximum operating temperature activates the Thermal Control Circuit (TCC). When activated the TCC causes both the processor IA core and graphics core to reduce frequency and voltage adaptively. The Adaptive Thermal Monitor will remain active as long as the package temperature remains at its specified limit. Therefore, the Adaptive Thermal Monitor will continue to reduce the package frequency and voltage until the TCC is de-activated.

$T_{jMAX}$  is factory calibrated and is not user configurable. The default value is software visible in the TEMPERATURE\_TARGET (0x1A2) MSR, bits [23:16].

The Adaptive Thermal Monitor does not require any additional hardware, software drivers, or interrupt handling routines. It is not intended as a mechanism to maintain processor thermal control to PL1 = TDP. The system design should provide a thermal solution that can maintain normal operation when PL1 = TDP within the intended usage range.

Adaptive Thermal Monitor protection is always enabled.

#### TCC Activation Offset

TCC Activation Offset can be set as an offset from  $T_{jMAX}$  to lower the onset of TCC and Adaptive Thermal Monitor. In addition, there is an optional time window (Tau) to manage processor performance at the TCC Activation offset value via an EWMA (Exponential Weighted Moving Average) of temperature.

#### TCC Activation Offset with Tau=0

An offset (degrees Celsius) can be written to the TEMPERATURE\_TARGET (0x1A2) MSR, bits [29:24], the offset value will be subtracted from the value found in bits [23:16]. When the time window (Tau) is set to zero, there will be no averaging, the offset, will be subtracted from the  $T_{jMAX}$  value and used as a new maximum

temperature set point for Adaptive Thermal Monitoring. This will have the same behavior as in prior products to have TCC activation and Adaptive Thermal Monitor to occur at this lower target silicon temperature.

If enabled, the offset should be set lower than any other passive protection such as ACPI\_PSV trip points

### **TCC Activation Offset with Tau**

To manage the processor with the EWMA (Exponential Weighted Moving Average) of temperature, an offset (degrees Celsius) is written to the TEMPERATURE\_TARGET (0x1A2) MSR, bits [29:24], and the time window (Tau) is written to the TEMPERATURE\_TARGET (0x1A2) MSR [6:0]. The offset value will be subtracted from the value found in bits [23:16] and be the temperature.

The processor will manage to this average temperature by adjusting the frequency of the various domains. The instantaneous  $T_j$  can briefly exceed the average temperature. The magnitude and duration of the overshoot is managed by the time window value (Tau).

This averaged temperature thermal management mechanism is in addition, and not instead of  $T_{jMAX}$  thermal management. That is, whether the TCC activation offset is 0 or not, TCC Activation will occur at  $T_{jMAX}$ .

### **Frequency / Voltage Control**

Upon Adaptive Thermal Monitor activation, the processor attempts to dynamically reduce processor temperature by lowering the frequency and voltage operating point. The operating points are automatically calculated by the processor IA core itself and do not require the BIOS to program them as with previous generations of Intel processors. The processor IA core will scale the operating points such that:

- The voltage will be optimized according to the temperature, the processor IA core bus ratio and the number of processor IA cores in deep C-states.
- The processor IA core power and temperature are reduced while minimizing performance degradation.

Once the temperature has dropped below the trigger temperature, the operating frequency and voltage will transition back to the normal system operating point.

Once a target frequency/bus ratio is resolved, the processor IA core will transition to the new target automatically.

- On an upward operating point transition, the voltage transition precedes the frequency transition.
- On a downward transition, the frequency transition precedes the voltage transition.
- The processor continues to execute instructions. However, the processor will halt instruction execution for frequency transitions.

If a processor load-based Enhanced Intel SpeedStep Technology/P-state transition (through MSR write) is initiated while the Adaptive Thermal Monitor is active, there are two possible outcomes:

- If the P-state target frequency is higher than the processor IA core optimized target frequency, the P-state transition will be deferred until the thermal event has been completed.

- If the P-state target frequency is lower than the processor IA core optimized target frequency, the processor will transition to the P-state operating point.

### **Clock Modulation**

If the frequency/voltage changes are unable to end an Adaptive Thermal Monitor event, the Adaptive Thermal Monitor will utilize clock modulation. Clock modulation is done by alternately turning the clocks off and on at a duty cycle (ratio between clock "on" time and total time) specific to the processor. The duty cycle is factory configured to 25% on and 75% off and cannot be modified. The period of the duty cycle is configured to 32 microseconds when the Adaptive Thermal Monitor is active. Cycle times are independent of processor frequency. A small amount of hysteresis has been included to prevent excessive clock modulation when the processor temperature is near its maximum operating temperature. Once the temperature has dropped below the maximum operating temperature, and the hysteresis timer has expired, the Adaptive Thermal Monitor goes inactive and clock modulation ceases. Clock modulation is automatically engaged as part of the Adaptive Thermal Monitor activation when the frequency/voltage targets are at their minimum settings. Processor performance will be decreased when clock modulation is active. Snooping and interrupt processing are performed in the normal manner while the Adaptive Thermal Monitor is active.

Clock modulation will not be activated by the Package average temperature control mechanism.

### **Thermal Throttling**

As the processor approaches  $T_{jMAX}$  a throttling mechanisms will engage to protect the processor from over-heating and provide control thermal budgets.

Achieving this is done by reducing IA and other subsystem agent's voltages and frequencies in a gradual and coordinated manner that varies depending on the dynamics of the situation. IA frequencies and voltages will be directed down as low as LFM (Lowest Frequency Mode). Further restricts are possible via Thermal Trolling point (TT1) under conditions where thermal budget cannot be re-gained fast enough with voltages and frequencies reduction alone. TT1 keeps the same processor voltage and clock frequencies the same yet skips clock edges to produce effectively slower clocking rates. This will effectively result in observed frequencies below LFM on the Windows PERF monitor.

#### **12.1.3.2 Digital Thermal Sensor**

Each processor has multiple on-die Digital Thermal Sensor (DTS) that detects the processor IA, GT and other areas of interest instantaneous temperature.

Temperature values from the DTS can be retrieved through:

- A software interface using processor Model Specific Register (MSR).
- A processor hardware interface.

When the temperature is retrieved by the processor MSR, it is the instantaneous temperature of the given DTS. When the temperature is retrieved using PECCI, it is the average of the highest DTS temperature in the package over a 256 ms time window. Intel recommends using the PECCI reported temperature for platform thermal control that benefits from averaging, such as fan speed control. The average DTS temperature



may not be a good indicator of package Adaptive Thermal Monitor activation or rapid increases in temperature that triggers the Out of Specification status bit within the PACKAGE\_THERM\_STATUS (0x1B1) MSR and IA32\_THERM\_STATUS (0x19C) MSR.

Code execution is halted in C1 or deeper C-states. Package temperature can still be monitored through PECI in lower C-states.

Unlike traditional thermal devices, the DTS outputs a temperature relative to the maximum supported operating temperature of the processor ( $T_{jMAX}$ ), regardless of TCC activation offset. It is the responsibility of software to convert the relative temperature to an absolute temperature. The absolute reference temperature is readable in the TEMPERATURE\_TARGET (0x1A2) MSR. The temperature returned by the DTS is an implied negative integer indicating the relative offset from  $T_{jMAX}$ . The DTS does not report temperatures greater than  $T_{jMAX}$ . Refer to the appropriate processor family BIOS Specification for specific register details. The DTS-relative temperature readout directly impacts the Adaptive Thermal Monitor trigger point. When a package DTS indicates that it has reached the TCC activation (a reading of 0x0, except when the TCC activation offset is changed), the TCC will activate and indicate an Adaptive Thermal Monitor event. A TCC activation will lower both processor IA core and graphics core frequency, voltage, or both. Changes to the temperature can be detected using two programmable thresholds located in the processor thermal MSRs. These thresholds have the capability of generating interrupts using the processor IA core's local APIC. Refer to the *Intel 64 Architectures Software Developer's Manual* for specific register and programming details.

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

The error associated with DTS measurements will not exceed  $\pm 5$  °C within the entire operating range.

#### Fan Speed Control with Digital Thermal Sensor

Digital Thermal Sensor based fan speed control ( $T_{FAN}$ ) is a recommended feature to achieve optimal thermal performance. At the  $T_{FAN}$  temperature, Intel recommends full cooling capability before the DTS reading reaches  $T_{jMAX}$ .

### 12.1.3.3 PROCHOT# Signal

The PROCHOT# (processor hot) signal is asserted by the processor when the TCC is active. Only a single PROCHOT# pin exists at a package level. When any DTS temperature reaches the TCC activation temperature, the PROCHOT# signal will be asserted. PROCHOT# assertion policies are independent of Adaptive Thermal Monitor enabling.

The PROCHOT# signal can be configured to the following modes:

- **Input Only:** PROCHOT is driven by an external device.
- **Output Only:** PROCHOT is driven by processor.
- **Bi-Directional:** Both Processor and external device can drive PROCHOT signal

#### PROCHOT Input Only

It is strongly recommended to have PROCHOT# signal set as **Input only** by default. In this state, the processor will only monitor PROCHOT# assertions and respond by setting the maximum frequency to 10kHz.

The following two features are enabled when PROCHOT is set to Input only:

- **Fast PROCHOT:** Respond to PROCHOT# within 1uS of PROCHOT# pin assertion, reducing the processor power.
- **PROCHOT Demotion Algorithm:** Designed to improve system performance during multiple PROCHOT assertions.

#### 12.1.3.4 PROCHOT Output Only

Legacy state, PROCHOT is driven by the processor to external device.

#### 12.1.3.5 Bi-Directional PROCHOT#

By default, the PROCHOT# signal is set to input only. When configured as an input or bi-directional signal, PROCHOT# can be used for thermally protecting other platform components should they overheat as well. When PROCHOT# is driven by an external device:

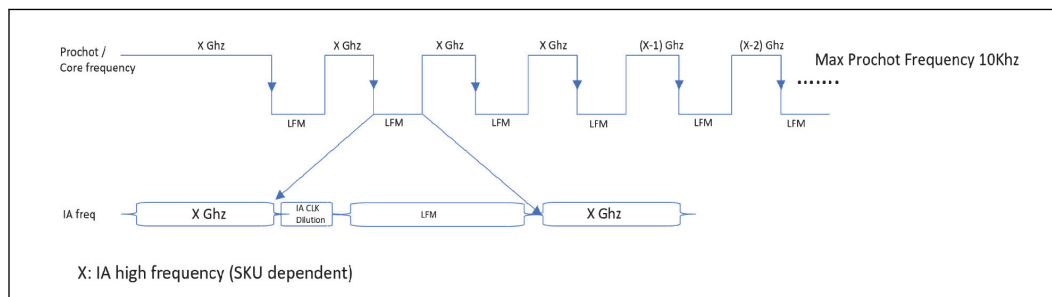
- The package will immediately transition to the lowest P-State (Pn) supported by the processor IA cores and graphics cores. This is contrary to the internally-generated Adaptive Thermal Monitor response.
- Clock modulation is not activated.

The processor package will remain at the lowest supported P-state until the system de-asserts PROCHOT#. The processor can be configured to generate an interrupt upon assertion and de-assertion of the PROCHOT# signal. Refer to appropriate processor family BIOS Specification for specific register and programming details.

When PROCHOT# is configured as a bi-directional signal and PROCHOT# is asserted by the processor, it is impossible for the processor to detect a system assertion of PROCHOT#. The system assertion will have to wait until the processor de-asserts PROCHOT# before PROCHOT# action can occur due to the system assertion. While the processor is hot and asserting PROCHOT#, the power is reduced but the reduction rate is slower than the system PROCHOT# response of < 100 us. The processor thermal control is staged in smaller increments over many milliseconds. This may cause several milliseconds of delay to a system assertion of PROCHOT# while the output function is asserted.

#### 12.1.3.6 PROCHOT Demotion Algorithm

PROCHOT demotion algorithm is designed to improve system performance following multiple Platform PROCHOT consecutive assertions. During each PROCHOT assertion processor will eventually transition to the lowest P-State (Pn) supported by the processor IA cores and graphics cores (LFM). When detecting several PROCHOT consecutive assertions the processor will reduce the max frequency in order to reduce the PROCHOT assertions events. The processor will keep reducing the frequency until no consecutive assertions detected. The processor will raise the frequency if no consecutive PROCHOT assertion events will occur. PROCHOT demotion algorithm enabled only when the PROCHOT is configured as input.

**Figure 14. PROCHOT Demotion Signal Description**

### 12.1.3.7 Voltage Regulator Protection using PROCHOT#

PROCHOT# may be used for thermal protection of voltage regulators (VR). System designers can create a circuit to monitor the VR temperature and assert PROCHOT# and, if enabled, activate the TCC when the temperature limit of the VR is reached. When PROCHOT# is configured as a bi-directional or input only signal, if the system assertion of PROCHOT# is recognized by the processor, results in power reduction as described in the section PROCHOT# Signal. Power reduction down to LFM and duration of the platform PROCHOT# assertion as described in [PROCHOT Demotion Algorithm](#) on page 138. supported by the processor IA cores and graphics cores. Systems should still provide proper cooling for the VR and rely on bi-directional PROCHOT# only as a backup in case of system cooling failure. Overall, the system thermal design should allow the power delivery circuitry to operate within its temperature specification even while the processor is operating at its TDP.

### 12.1.3.8 Thermal Solution Design and PROCHOT# Behavior

With a properly designed and characterized thermal solution, it is anticipated that PROCHOT# will only be asserted for very short periods of time when running the most power intensive applications. The processor performance impact due to these brief periods of TCC activation is expected to be so minor that it would be immeasurable. However, an under-designed thermal solution that is not able to prevent excessive assertion of PROCHOT# in the anticipated ambient environment may:

- Cause a noticeable performance loss.
- Result in prolonged operation at or above the specified maximum junction temperature and affect the long-term reliability of the processor.
- May be incapable of cooling the processor even when the TCC is active continuously (in extreme situations).

### 12.1.3.9 Low-Power States and PROCHOT# Behavior

Depending on package power levels during package C-states, outbound PROCHOT# may de-assert while the processor is idle as power is removed from the signal. Upon wake up, if the processor is still hot, the PROCHOT# will re-assert, although typically package idle state residency should resolve any thermal issues. The PECCI interface is fully operational during all C-states and it is expected that the platform continues to manage processor IA core and package thermals even during idle states by regularly polling for thermal data over PECCI.

### 12.1.3.10 THRMTRIP# Signal

Regardless of enabling the automatic or on-demand modes, in the event of a catastrophic cooling failure, the package will automatically shut down when the silicon has reached an elevated temperature that risks physical damage to the product. At this point, the THRMTRIP# signal will go active.

### 12.1.3.11 Critical Temperature Detection

Critical Temperature Detection is performed by monitoring the package temperature. This feature is intended for graceful shutdown before the THRMTRIP# is activated. However, the processor execution is not guaranteed between critical temperature and THRMTRIP#. If the Adaptive Thermal Monitor is triggered and the temperature remains high, a critical temperature status and sticky bit are latched in the PACKAGE\_THERM\_STATUS (0x1B1) MSR and the condition also generates a thermal interrupt, if enabled. For more details on the interrupt mechanism, refer to Intel® 64 Architectures Software Developer's Manual or appropriate processor family BIOS Specification.

### 12.1.3.12 On-Demand Mode

The processor provides an auxiliary mechanism that allows system software to force the processor to reduce its power consumption using clock modulation. This mechanism is referred to as "On-Demand" mode and is distinct from Adaptive Thermal Monitor and bi-directional PROCHOT#. The processor platforms should not rely on software usage of this mechanism to limit the processor temperature. On-Demand Mode can be accomplished using processor MSR or chipset I/O emulation. On-Demand Mode may be used in conjunction with the Adaptive Thermal Monitor. However, if the system software tries to enable On-Demand mode at the same time the TCC is engaged, the factory configured the duty cycle of the TCC will override the duty cycle selected by the On-Demand mode. If the I/O based and MSR-based On-Demand modes are in conflict, the duty cycle selected by the I/O emulation-based On-Demand mode will take precedence over the MSR-based On-Demand Mode. For more details, refer to appropriate processor family BIOS Specification.

### 12.1.3.13 MSR Based On-Demand Mode

If Bit 4 of the IA32\_CLOCK\_MODULATION MSR is set to 1, the processor will immediately reduce its power consumption using modulation of the internal processor IA core clock, independent of the processor temperature. The duty cycle of the clock modulation is programmable using bits [3:1] of the same IA32\_CLOCK\_MODULATION MSR. In this mode, the duty cycle can be programmed in either 12.5% or 6.25% increments (discoverable using CPUID). Thermal throttling using this method will modulate each processor IA core's clock independently. For more details, refer to the appropriate processor family BIOS Specification.

### 12.1.3.14 I/O Emulation-Based On-Demand Mode

I/O emulation-based clock modulation provides legacy support for operating system software that initiates clock modulation through I/O writes to ACPI defined processor clock control registers on the chipset (PROC\_CNT). Thermal throttling using this method will modulate all processor IA cores simultaneously. For more details, refer to appropriate processor family BIOS Specification Section.

## 12.1.4 Intel® Memory Thermal Management

### DRAM Thermal Aggregation

P-Unit firmware is responsible for aggregating DRAM temperature sources into a per-DIMM reading as well as an aggregated virtual 'max' sensor reading. At reset, MRC communicates to the MC the valid channels and ranks as well as DRAM type. At that time, Punit firmware sets up a valid channel and rank mask that is then used in the thermal aggregation algorithm to produce a single maximum temperature.

### DRAM Thermal Monitoring

- DRAM thermal sensing Periodic DDR thermal reads from DDR.
- DRAM thermal calculation Punit reads of DDR thermal information direct from the memory controller (MR4 or MPR) Punit estimation of a virtual maximum DRAM temperature based on per-rank readings. Application of thermal filter to the virtual maximum temperature.

### DRAM Refresh Rate Control

The MRC will natively interface with MR4 or MPR readings to adjust DRAM refresh rate as needed to maintain data integrity. This capability is enabled by default and occurs automatically. Direct override of this capability is available for debug purposes, but this cannot be adjusted during runtime.

### DRAM Bandwidth Throttling (Change to DDR Bandwidth Throttling)

Control for bandwidth throttling is available through the memory controller. Software may program a percentage bandwidth target at the current operating frequency and that used to throttle read and write commands based on the maximum memory MPR/MR4 reading.

## 12.2 Processor Line Thermal and Power Specifications

The following notes apply to [Processor Line Thermal and Power](#) on page 142

Note	Definition
1	The Processor Base Power (a.k.a TDP) and Assured Power (cTDP) values are the average power dissipation in junction temperature operating condition limit, for the SKU Segment and Configuration, for which the processor is validated during manufacturing when executing an associated Intel-specified high-complexity workload at the processor IA core frequency corresponding to the configuration and SKU.
2	Processor Base Power (a.k.a TDP) workload may consist of a combination of processor IA core intensive and graphics core intensive applications.
3	Can be modified at runtime by MSR writes, with MMIO and with PECI commands.
4	'Turbo Time Parameter' is a mathematical parameter (units of seconds) that controls the processor turbo algorithm using a moving average of energy usage. Do not set the Turbo Time Parameter to a value less than 0.1 seconds. refer to <a href="#">Platform Power Control</a> on page 131 for further information.
5	The shown limit is a time averaged-power, based upon the Turbo Time Parameter. Absolute product power may exceed the set limits for short durations or under virus or uncharacterized workloads.
6	The Processor will be controlled to a specified power limit. If the power value and/or 'Turbo Time Parameter' is changed during runtime, it may take a short period of time (approximately 3 to 5 times the 'Turbo Time Parameter') for the algorithm to settle at the new control limits.
7	This is a hardware default setting and not a behavioral characteristic of the part.

*continued...*

Note	Definition
8	For controllable turbo workloads, the PL2 limit may be exceeded for up to 10ms.
9	LPM power level is an opportunistic power and is not a guaranteed value as usages and implementations may vary.
10	Power limits may vary depending on if the product supports the 'Maximum Assured Power (cTDP Up)' and/or 'Minimum Assured Power(cTDP Down)' modes. Default power limits can be found in the PKG_PWR_SKU MSR (614h).
11	The processor die do not reach maximum sustained power simultaneously since the sum of the 2 die's estimated power budget is controlled to be equal to or less than the package Processor Base Power (a.k.a TDP) (PL1) limit.
12	Minimum Assured Power(cTDP Down) power is based on 32EU equivalent graphics configuration. Minimum Assured Power(cTDP Down) down does not decrease the number of active Processor Graphics EUs but relies on Power Budget Management (PL1) to achieve the specified power level.
13	May vary based on SKU.
14	<ul style="list-style-type: none"> <li>The formula of <math>PL2=PL1*1.25</math> is the hardware.</li> <li>PL2- Processor opportunistic higher Average Power with limited duration controlled by Tau_PL1 setting, the larger the Tau, the longer the PL2 duration.</li> </ul>
15	Processor Base Power (a.k.a TDP) workload does not reflect various I/O connectivity cases.
16	Hardware default of PL1 Tau=1s, By including the benefits available from power and thermal management features the recommended is to use PL1 Tau=28s.

## 12.2.1 Processor Line Thermal and Power

Table 53. Package Turbo Specifications

Processor IA Cores, Graphics, Configuration and Processor Base Power (a.k.a TDP)	Parameter	Minimum	Tau MSR Max Value	Recommended Value	Units	Notes
8 E-Cores 15W	Power Limit 1 Time (PL1 Tau)	0.1	448	28	S	3,4,5,6,7,8,14,16,17
	Power Limit 1 (PL1)	N/A	N/A	15	W	
	Power Limit 2 (PL2)	N/A	N/A	Note	W	
8 E-Cores 9W	Power Limit 1 Time (PL1 Tau)	0.1	448	28	S	3,4,5,6,7,8,14,16,17
	Power Limit 1 (PL1)	N/A	N/A	9	W	
	Power Limit 2 (PL2)	N/A	N/A	Note	W	
4 E-Cores 6W	Power Limit 1 Time (PL1 Tau)	0.1	448	28	S	3,4,5,6,7,8,14,16,17

*continued...*

Processor IA Cores, Graphics, Configuration and Processor Base Power (a.k.a TDP)	Parameter	Minimum	Tau MSR Max Value	Recommended Value	Units	Notes
	Power Limit 1 (PL1)	N/A	N/A	6	W	
	Power Limit 2 (PL2)	N/A	N/A	Note	W	

Notes:

- No Specifications for Min/Max PL1/PL2 values.
- Hardware default of PL1 Tau=1s, By including the benefits available from power and thermal management features the recommended is to use PL1 Tau=28s.
- PL2- Processor opportunistic higher Average Power – Reactive, Limited Duration controlled by Tau\_PL1 setting.
- PL1 Tau - PL1 average power is controlled via PID algorithm with this Tau, The larger the Tau, the longer the PL2 duration.
- System cooling solution and designs found to not being able to support the Performance TauPL1, adjust the TauPL1 to cooling capability.

**Table 54. Junction Temperature Specifications**

Segment	Symbol	Package Turbo Parameter	Temperature Range		Processor Base Power (a.k.a TDP) Specification Temperature Range		Units	Notes
			Minimum	Maximum	Minimum	Maximum		
BGA	Tj	Junction temperature limit	0	105	35	105	°C	1, 2

Notes:

- The thermal solution needs to ensure that the processor temperature does not exceed the Processor Base Power (a.k.a TDP) Specification Temperature Range.
- The processor junction temperature is monitored by Digital Temperature Sensors (DTS). For DTS accuracy, refer to [Digital Thermal Sensor](#) on page 136.

### 12.3 Error and Thermal Protection Signals

**Table 55. Error and Thermal Protection Signals**

Signal Name	Description	Dir.	Buffer Type	Link Type
CATERR#	<b>Catastrophic Error:</b> This signal indicates that the system has experienced a catastrophic error and cannot continue to operate. The processor will set this signal for non-recoverable machine check errors or other unrecoverable internal errors. CATERR# is used for signaling the following types of errors: Legacy MCERRs, CATERR# is asserted for 16 BCLKs. Legacy IERRs, CATERR# remains asserted until warm or cold reset.	O	OD	SE
PECI	<b>Platform Environment Control Interface:</b> A serial sideband interface to the processor. It is used primarily for thermal, power, and error management. Details regarding the Peci electrical specifications, protocols and functions can be found in the RS-Platform Environment Control Interface (PECI) Specification, Revision 3.0.	I/O	PECI, Async	SE
PROCHOT#	<b>Processor Hot:</b> PROCHOT# goes active when the processor temperature monitoring sensor(s) detects that the processor has reached its maximum safe operating temperature. This	I/O	I:GTL/ O:OD	SE

*continued...*

Signal Name	Description	Dir.	Buffer Type	Link Type
	indicates that the processor Thermal Control Circuit (TCC) has been activated, if enabled. This signal can also be driven to the processor to activate the TCC.			
THERMTRIP#	<b>Thermal Trip:</b> The processor protects itself from catastrophic overheating by use of an internal thermal sensor. This sensor is set well above the normal operating temperature to ensure that there are no false trips. The processor will stop all executions when the junction temperature exceeds approximately 130 °C. This is signaled to the system by the THRMTRIP# pin.	O	OD	SE



## 13.0 PCH Thermal Sensor

---

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

### 13.1 Modes of Operation

The DTS has two usages when enabled:

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

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

### 13.2 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.

### 13.3 Thermal Sensor Accuracy ( $T_{\text{accuracy}}$ )

The PCH thermal sensor accuracy is:

- $\pm 5$  °C over the temperature range from 50 °C to 110 °C.
- $\pm 7$  °C over the temperature range from 30 °C to 50 °C.
- $\pm 10$  °C over the temperature range from -10 °C to 30 °C.
- No accuracy is specified for temperature range beyond 110 °C or below -10 °C.

## 13.4 Thermal Reporting to an EC

To support a platform EC that is managing the system thermals, the PCH provides the ability for the EC to read the PCH temperature over SMBus and/or over eSPI. If enabled, PMC will drive the temperature directly to the SMBus and eSPI units. The EC will issue an SMBus read or eSPI OOB Channel request and receives a single byte of data, indicating a temperature between 0°C and 127°C, where 255 (0xFF) indicates that the sensor isn't enabled yet. The EC must be connected to either SMLink1 or eSPI for thermal reporting support.

---

### NOTE

The catastrophic trip value is set to 120 °C and is not programmed or accessible by BIOS.

---

## 13.5 Thermal Trip Signal (PCHHOT#)

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

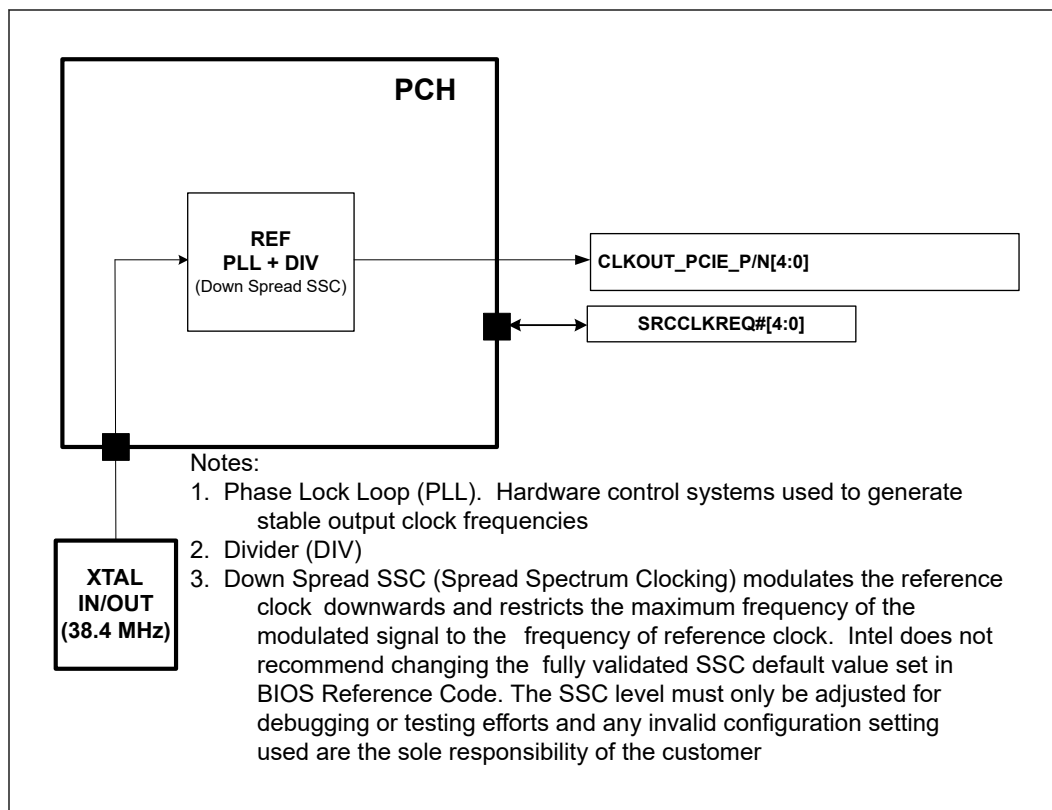
PCHHOT# is an O/D output and requires a Pull-up on the motherboard.

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

## 14.0 System Clocks

### 14.1 ICC

Figure 15. Integrated Clock Controller (ICC) Diagram



#### 14.1.1 Signal Descriptions

Table 56. Signal Descriptions

Name	Type	SSC Capable	Description
CLKOUT_PCIE_N0 CLKOUT_PCIE_N1 CLKOUT_PCIE_N2 CLKOUT_PCIE_N3 CLKOUT_PCIE_N4 / UFS_REF_CLK CLKOUT_PCIE_P0 CLKOUT_PCIE_P1 CLKOUT_PCIE_P2	O	Yes	<b>PCI Express* Clock Output:</b> Serial Reference 100 MHz PCIe* specification compliant differential output clocks to PCIe* devices
			<i>continued...</i>

Name	Type	SSC Capable	Description
<b>CLKOUT_PCIE_P3</b> <b>CLKOUT_PCIE_P4</b>			
GPP_D4 / <b>IMGCLKOUT0</b> / BK4 / SBK4	O		<b>Imaging Clock</b> : Clock for external camera sensor
GPP_H20 / <b>IMGCLKOUT1</b>	O		<b>Imaging Clock</b> : Clock for external camera sensor
GPP_H21 / <b>IMGCLKOUT2</b>	O		<b>Imaging Clock</b> : Clock for external camera sensor
GPP_H22 / <b>IMGCLKOUT3</b>	O		<b>Imaging Clock</b> : Clock for external camera sensor
GPP_D5 / <b>SRCLKREQ0#</b> GPP_D6 / <b>SRCLKREQ1#</b> GPP_D7 / <b>SRCLKREQ2#</b> GPP_D8 / <b>SRCLKREQ3#</b> GPP_H19 / <b>SRCLKREQ4#</b>	I/O		<b>Clock Request</b> : Serial Reference Clock request signals for PCIe* 100 MHz differential clocks
<b>XTAL_IN</b>	I		<b>Crystal Input</b> : Input connection for 38.4 MHz crystal to PCH
<b>XTAL_OUT</b>	O		<b>Crystal Output</b> : Output connection for 38.4 MHz crystal to PCH
<b>XCLK_BIASREF</b>	I/O		<b>Differential Clock Bias Reference</b> : Used to set BIAS reference for differential clocks
<p><i>Notes:</i> 1. SSC = Spread Spectrum Clocking. Intel does not recommend changing the Plan of Record and fully validated SSC default value set in BIOS Reference Code. The SSC level must only be adjusted for debugging or testing efforts and any invalid configuration setting used are the sole responsibility of the customer.</p> <p>2. The SRCLKREQ# signals can be configured to map to any of the PCH PCI Express* Root Ports while using any of the CLKOUT_PCIE_P/N differential pairs</p>			

## 14.2 I/O Signal Pin States

**Table 57. I/O Signal Pin States**

Signal Name	S3/S4/S5	S0 Entry	S0	Deep Sx
CLKOUT_PCIE_P[0:4] CLKOUT_PCIE_N[0:4]	OFF (Gated Low)	Bringing up the Clock	Toggling	OFF (Gated Low)
SRCLKREQ[0:4]#	Un-driven	Un-driven	Driven	OFF

## 14.3 Clock Topology

The processor has 2 reference clocks that drive the various components within the Processor:

- Processor reference clock or base clock (BCLK). 100MHz with SSC.
- Fixed clock. 38.4MHz without SSC (crystal clock).

BCLK drives the following clock domains:

- Core
- Ring
- Graphics (GT)

- Memory Controller (MC)
- System Agent (SA)

PCTGLK drives the following clock domains:

- OPIO

Fixed clock drives the following clock domains:

- Display
- SVID controller
- Time Stamp Counters (TSC)
- Type C subsystem

### **14.3.1 Integrated Reference Clock PLL**

The processor includes a phase lock loop (PLL) that generates the reference clock for the processor from a fixed crystal clock. The processor reference clock is also referred to as Base Clock or BCLK.

By integrating the BCLK PLL into the processor die, a cleaner clock is achieved at a lower power compared to the legacy PCH BCLK PLL solution.

The BCLK PLL has controls for RFI/EMI mitigations.

## 15.0 Real Time Clock (RTC)

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

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

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

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

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

**Table 58. Acronyms**

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

### 15.1 Signal Description

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

*continued...*

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

## 15.2 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>RTCRST#</b>	RTC	Undriven	Undriven	Undriven	Undriven
<b>SRTCST#</b>	RTC	Undriven	Undriven	Undriven	Undriven

*Note:* 1. Reset reference for RTC well pins is RTCRST#.

## 16.0 Memory

### 16.1 Signal Description

Table 59. DDR4 Memory Interface

Signal Name	Description	Dir.	Buffer Type	Link Type
DDR0_DQ[7:0][[7:0]]	<b>Data Buses:</b> Data signals interface to the SDRAM data buses. <b>Example:</b> DDR0_DQ2[5] refers to DDR channel 0, Byte 2, Bit 5.	I/O	DDR4	SE
DDR0_DQSP[7:0] DDR0_DQSN[7:0]	<b>Data Strobes:</b> Differential data strobe pairs. The data is captured at the crossing point of DQS during reading and write transactions. <b>Example:</b> DDR0_DQSP0 refers to DQSP of DDR channel 0, Byte 0.	I/O	DDR4	Diff
DDR0_CLKN[1:0] DDR0_CLKP[1:0]	<b>SDRAM Differential Clock:</b> Differential clocks signal pairs, pair per rank. The crossing of the positive edge and the negative edge of their complement are used to sample the command and control signals on the SDRAM.	O	DDR4	Diff
DDR0_CKE[1:0]	<b>Clock Enable:</b> (1 per rank). These signals are used to: <ul style="list-style-type: none"> <li>• Initialize the SDRAMs during power-up.</li> <li>• Power-down SDRAM ranks.</li> <li>• Place all SDRAM ranks into and out of self-refresh during STR (Suspend to RAM).</li> </ul>	O	DDR4	SE
DDR0_CS[1:0]	<b>Chip Select:</b> (1 per rank). These signals are used to select particular SDRAM components during the active state. There is one Chip Select for each SDRAM rank.	O	DDR4	SE
DDR0_ODT[1:0]	<b>On Die Termination:</b> (1 per rank). Active SDRAM Termination Control.	O	DDR4	SE
DDR0_MA[16:0]	<b>Address:</b> These signals are used to provide the multiplexed row and column address to the SDRAM. DDR0_MA[16] uses as RAS# signal DDR0_MA[15] uses as CAS# signal DDR0_MA[14] uses as WE# signal	O	DDR4	SE
DDR0_ACT_N	<b>Activation Command:</b> ACT# HIGH along with CS_N determines that the signals addresses below have command functionality.	O	DDR4	SE
DDR0_BG[1:0]	<b>Bank Group:</b> BG[1:0] define to which bank group an Active, reading, Write or Precharge command is being applied. BG0 also determines which mode register is to be accessed during a MRS cycle.	O	DDR4	SE

*continued...*



Signal Name	Description	Dir.	Buffer Type	Link Type
DDR0_BA[1:0]	<b>Bank Address:</b> BA[1:0] define to which bank an Active, reading, Write or Precharge command is being applied. Bank address also determines which mode register is to be accessed during a MRS cycle.	O	DDR4	SE
DDR0_VREF_CA0	<b>Memory Reference Voltage for Command and Address</b>	O	A	SE
DDR_VTT_CTL	<b>System Memory Power Gate Control:</b> When signal is high – platform memory VTT regulator is enable, output high. When signal is low - Disables the platform memory VTT regulator in C8 and deeper and S3.	O	A	SE

Table 60. LP5 Memory Interface

Signal Name	Description	Dir.	Buffer Type	Link Type
DDR0_DQ[1:0][7:0] DDR1_DQ[1:0][7:0] DDR2_DQ[1:0][7:0] DDR3_DQ[1:0][7:0]	<b>Data Buses:</b> Data signals interface to the SDRAM data buses. <b>Example:</b> DDR0_DQ1[5] refers to DDR channel 0, Byte 1, Bit 5.	I/O	LP5	SE
DDR0_DQSP[1:0] DDR1_DQSP[1:0] DDR2_DQSP[1:0] DDR3_DQSP[1:0] DDR0_DQSN[1:0] DDR1_DQSN[1:0] DDR2_DQSN[1:0] DDR3_DQSN[1:0]	<b>Data Strobes:</b> Differential data strobe pairs. The data is captured at the crossing point of DQS during reading and write transactions.	I/O	LP5	Diff
DDR0_CLK_N DDR0_CLK_P DDR1_CLK_N DDR1_CLK_P DDR2_CLK_N DDR2_CLK_P DDR3_CLK_N DDR3_CLK_P	<b>SDRAM Differential Clock:</b> Differential clocks signal pairs, pair per channel and package. The crossing of the positive edge and the negative edge of their complement are used to sample the command and control signals on the SDRAM.	I/O	LP5	Diff
DDR0_CS[1:0] DDR1_CS[1:0] DDR2_CS[1:0] DDR3_CS[1:0]	<b>Chip Select:</b> (1 per rank). These signals are used to select particular SDRAM components during the active state. There is one Chip Select for each SDRAM rank. The Chip select signal is Active High.	I/O	LP5	SE
DDR0_CA[5:0] DDR1_CA[5:0] DDR2_CA[5:0] DDR3_CA[5:0]	<b>Command Address:</b> These signals are used to provide the multiplexed command and address to the SDRAM.	I/O	LP5	SE

continued...

Signal Name	Description	Dir.	Buffer Type	Link Type
DDR[3:0]_WCK_P DDR[3:0]_WCK_N	<b>Write Clocks:</b> WCK_N and WCK_P are differential clocks used for WRITE data capture and READ data output.	O	LP5	Diff
DDR_COMP	<b>System Memory Resistance Compensation</b>	A	A	SE
DRAM_RESET#	<b>Memory Reset</b>	O	CMOS	SE

**Table 61. DDR5 Memory Interface**

Signal Name	Description	Dir.	Buffer Type	Link Type
DDR0_DQ[3:0][[7:0]] DDR1_DQ[3:0][[7:0]]	<b>Data Buses:</b> Data signals interface to the SDRAM data buses. <b>Example:</b> DDR0_DQ2[5] refers to DDR channel 0, Byte 2, Bit 5.	I/O	DDR5	SE
DDR0_DQSP[3:0] DDR0_DQSN[3:0] DDR1_DQSP[3:0] DDR1_DQSN[3:0]	<b>Data Strobes:</b> Differential data strobe pairs. The data is captured at the crossing point of DQS during reading and write transactions. <b>Example:</b> DDR0_DQSP0 refers to DQSP of DDR channel 0, Byte 0.	O	DDR5	Diff
DDR0_CLKN[1:0] DDR0_CLKP[1:0] DDR1_CLKN[1:0] DDR1_CLKP[1:0]	<b>SDRAM Differential Clock:</b> Differential clocks signal pairs, pair per rank. The crossing of the positive edge and the negative edge of their complement are used to sample the command and control signals on the SDRAM.	O	DDR5	Diff
DDR0_CS[1:0] DDR1_CS[1:0]	<b>Chip Select:</b> (1 per rank). These signals are used to select particular SDRAM components during the active state. There is one Chip Select for each SDRAM rank. The Chip select signal is Active High.	O	DDR5	SE
DDR0_CA[12:0] DDR1_CA[12:0]	<b>Command Address:</b> These signals are used to provide the multiplexed command and address to the SDRAM.	O	DDR5	SE

## 16.2 System Memory Interface

### 16.2.1 DDR Support Matrix

**Table 62. DDR Support Matrix Table**

Technology	DDR4	DDR5	LPDDR5 <sup>7</sup>
<b>Configuration</b>	1DPC	1DPC	1R/2R
<b>Maximum Frequency [MT/s]</b>	3200	4800	Type3: 4800
<b>VDDQ [V] <sup>5</sup></b>	1.2	5,1.1 <sup>7</sup>	0.5
<b>VDD2 [V] <sup>5</sup></b>	1.2	1.1	1.05
<i>continued...</i>			

Technology	DDR4	DDR5	LPDDR5 <sup>7</sup>
Maximum RPC <sup>2</sup>	2	2	2
Die Density [Gb]	8,16	16	8,12, 16

Notes: 1. 1DPC refer to when only 1DIMM slot per channel is routed.  
2. RPC = Rank Per Channel  
3. Memory down of all technologies should be implemented homogeneous means that all DRAM devices should be from the same vendor and have the same part number. Implementing a mix of DRAM devices may cause serious signal integrity and functional issues.  
4. There is no support for memory modules with different technologies or capacities on opposite sides of the same memory module. If one side of a memory module is populated, the other side is either identical or empty.  
5. VDD2 is Processor and DRAM voltage, and VDDQ is DRAM voltage.  
6. 5V is SoDIMM/UDIMM voltage, 1.1V is Memory down voltage.  
7. LP5x memory combo devices in LP5-8B mode is supported.

**Table 63. DDR Technology Support Matrix**

Technology	Form Factor	Ball Count
DDR4	SoDIMM	260
DDR4	x16 SDP (1R) <sup>1</sup>	96
DDR4	x16 DDP (1R) <sup>1</sup>	96
DDR4	x8 SDP (1R) <sup>1</sup>	78
DDR5	SoDIMM	262
DDR5	x8 SDP (1R) <sup>1</sup>	78
DDR5	x16 SDP (1R) <sup>1</sup>	102
LPDDR5	x32 (1R, 2R) <sup>1</sup>	315

**NOTE**

Memory down of all technologies should be implemented homogeneously, which means that all DRAM devices should be from the same vendor and have the same part number. Implementing a mix of DRAM devices may cause serious signal integrity and functional issues (all DRAMs in the system must be from same Part Number).

## 16.2.2 Supported Memory Modules and Devices

**Table 64. Supported DDR4 Non-ECC SoDIMM Module Configurations**

Raw Card Version	DIMM Capacity	DRAM Device Technology	DRAM Organization	# of DRAM Devices	# of Ranks	# of Row/Col Address Bits	# of Banks Inside DRAM	Page Size
A	8 GB	8 Gb	1024M x 8	8	1	16/10	16	8K
A	16 GB	16 Gb	2048M x 8	8	1	17/10	16	8K
C	4 GB	8 Gb	512M x 16	4	1	16/10	8	8K
C	8 GB	16 Gb	1024M x 16	4	1	17/10	8	8K
E	16 GB	8 Gb	1024M x 8	16	2	16/10	16	8K

**Table 65. Supported DDR5 Non-ECC SoDIMM Module Configurations**

Raw Card Version	DIMM Capacity	DRAM Device Technology	DRAM Organization	# of DRAM Devices	# of Ranks	# of Row/Col Address Bits	# of Banks Inside DRAM	Page Size
A	16 GB	16 Gb	2048M x 8	8	1	17/10	16	8K
C	8 GB	16 Gb	1024M x 16	4	1	17/10	8	8K

**Table 66. Supported DDR4 Memory Down Device Configurations**

Maximum System Capacity	PKG Type (Die bits x Package bits)	DRAM Organization / Package Type	Package Density	Die Density	Dies Per Channel	Rank Per Channel	PKGs Per Channel	Physical Device Rank	Banks Inside DRAM	Page Size
16 GB	SDP 8x8	1024M x 8	8 Gb	8 Gb	16	2	16	1	16	8K
4 GB	SDP 16x16	512M x 16	8 Gb	8 Gb	4	1	4	1	8	8K
8 GB <sup>1</sup>	SDP 16x16	1024M x 16	16 Gb	16 Gb	4	1	4	1	8	8K
8 GB	DDP 8x16	1024M x 16	16 Gb	8 Gb	8	1	4	1	16	8K
16 GB <sup>2</sup>	DDP 8x16	2048M x 16	32 Gb	16 Gb	8	1	4	1	16	8K

Notes: 1. For SDP: 1Rx16 using 16 Gb die density - the maximum system capacity is 8 GB  
 2. For DDP: 1Rx16 using 16 Gb die density - the maximum system capacity is 16 GB.

**Table 67. Supported DDR5 Memory Down Device Configurations**

Maximum System Capacity	PKG Type (Die bits x Package bits)	DRAM Organization / Package Type	Package Density	Die Density	Dies Per Channel	Rank Per Channel	PKGs Per channel	Physical Device Rank	Banks Inside DRAM	Page Size
16 GB	SDP 8x8	2048M x 8	16 Gb	16 Gb	8	1	8	1	16	8K
8 GB <sup>1</sup>	SDP 16x16	1024M x 16	16 Gb	16 Gb	4	1	4	1	8	8K

Note: 1. For SDP: 1Rx16 using 16 Gb die density - the maximum system capacity is 8 GB

**Table 68. Supported LPDDR5 x32 DRAMs Configurations**

Maximum System Capacity <sup>4</sup>	PKG Type	(Die bits per Ch x PKG bits) <sup>2</sup>	Die Density	PKG Density	Rank Per PKGs
8 GB	DDP	16x32	16 Gb	4 GB	1
6 GB	DDP	16x32	12 Gb	3 GB	1
16 GB	QDP	16x32	16 Gb	8 GB	2
4 GB	DDP	16x32	8 Gb	2 GB	1
8 GB	QDP	16x32	8 Gb	4 GB	2

Notes: 1. x32 BGA devices are 315 balls  
 2. DDP = Dual Die Package, QDP = Quad Die Package  
 3. Each LPDDR5 channel include two sub-channels  
 4. Maximum system capacity refers to system with all 4 sub-channels populated  
 5. Supports 8 bank mode (8B) LPDDR5 only.

**Table 69. Supported LPDDR5 x64 DRAMs Configurations**

Maximum System Capacity <sup>4</sup>	PKG Type	(Die bits per Ch x PKG bits) <sup>2</sup>	Die Density	PKG Density	DRAM Channels Per PKGs	Rank Per PKGs
8 GB <sup>1</sup>	QDP	16x64	16 Gb	8 GB	4	1
16 GB <sup>1</sup>	ODP	16x64	16 Gb	16 GB	4	2
4 GB <sup>1</sup>	QDP	16x64	8 Gb	4 GB	4	1
8 GB <sup>1</sup>	ODP	16x64	8 Gb	8 GB	4	2

*Notes:* 1. LP5 x64 DRAM (496b) will be supported as a WP config.  
2. QDP = Quad Die Package, ODP-Octal Die Package  
3. Maximum system capacity refers to systems with all 4 sub-channels populated

### 16.2.3 System Memory Timing Support

The IMC supports the following DDR Speed Bin, CAS Write Latency (CWL), and command signal mode timings on the main memory interface:

- tCL = CAS Latency
- tRCD = Activate Command to READ or WRITE Command delay
- tRP = PRECHARGE Command Period
- tRPb = per-bank PRECHARGE time
- tRPab = all-bank PRECHARGE time
- CWL = CAS Write Latency
- Command Signal modes:
  - 2N indicates a new DDR5/DDR4/LPDDR5 command may be issued every 2 clocks
  - 1N indicates a new DDR5/DDR4/LPDDR5 command may be issued every clock.

**Table 70. DDR System Memory Timing Support**

DRAM Device	Transfer Rate (MT/s)	tCL (tCK)	tRCD (ns)	tRP (ns)	CWL (tCK)	DPC	CMD Mode
DDR4	3200	22	13.75	13.75	9-12, 14,16,18,20	1,2	2N
DDR5	4000	36	17	17.00	34	1	2N
	4400	40	16.82	16.82	38	1,2 <sup>1</sup>	2N
	4800	40	16.67	16.67	38	1	2N

*Note:*  
1. 2 DPC supported when one slot is populated in each channel

**Table 71. LPDDR System Memory Timing Support**

DRAM Device	Transfer Rate (MT/s)	tCL (tCK)	tRCD (ns)	tRPpb (ns)	tRPab (ns)	WL (tCK) Set B
LPDDR5	4800	13	18.33	18.33	21.67	12

### 16.2.4 SAGV Points

SAGV (System Agent Geyserville) is a way by which the SoC can dynamically scale the work point (V/F), by applying DVFS (Dynamic Voltage Frequency Scaling) based on memory bandwidth utilization and/or the latency requirement of the various workloads for better energy efficiency at System-Agent. Pcode heuristics are in charge of providing request for Qclock work points by periodically evaluating the utilization of the memory and IA stalls.

**Table 72. SA Speed Enhanced Speed Steps (SA-GV) and Gear Mode Frequencies**

Technology	DDR Maximum Rate [MT/s]	SAGV-LowBW	SAGV-MedBW	SAGV-HighBW	SAGV- High Performance
LPDDR5 (7W/15W)	4800	2400 G4	4400 G4	4800 G4	4800 G2
LPDDR5 (6W)	4800	2400 G4	4000 G4	4400 G4	4800 G4
DDR4 (7W/15W/6W)	3200	2133 G2	2667 G2	2933 G2	3200 G2
DDR5 (7W/15W)	4800	2000 G2	4400 G4	4800 G4	4800 G2
DDR5 (6W)	4800	2000 G2	4000 G4	4400 G4	4800 G4

*Notes:* 1. The processors supports dynamic gearing technology where the Memory Controller can run at 1:1 (Gear-1, Legacy mode) or 1:2 (Gear-2 mode) and 1:4 (Gear-4 mode) ratio of DRAM speed. The gear ratio is the ratio of DRAM speed to Memory Controller Clock. MC Channel Width equal to DDR Channel width multiply by Gear Ratio

2. SA-GV modes

- LowBW** - Low frequency point, Minimum Power point. Characterized by low power, low BW, high latency. The system will stay at this point during low to moderate BW consumption.
- MedBW** - Tuned for balance between power and performance
- HighBW** - Characterized by high power, low latency, moderate BW also used as RFI mitigation point.
- MaxBW/Lowest Latency** - Lowest Latency point, low BW and highest power.

### 16.2.5 Memory Controller (MC)

The integrated memory controller is responsible for transferring data between the processor and the DRAM as well as the DRAM maintenance. The controller is capable of supporting up to four channels of LPDDR5 or two channels of DDR5 or one channel of DDR4.

### 16.2.6 Technology Enhancements of Intel® Fast Memory Access (Intel® FMA)

The following sections describe the Just-in-Time Scheduling, Command Overlap, and Out-of-Order Scheduling Intel FMA technology enhancements.

#### Just-in-Time Command Scheduling

The memory controller has an advanced command scheduler where all pending requests are examined simultaneously to determine the most efficient request to be issued next. The most efficient request is picked from all pending requests and issued to system memory Just-in-Time to make optimal use of Command Overlapping. Thus,

instead of having all memory access requests go individually through an arbitration mechanism forcing requests to be executed one at a time, they can be started without interfering with the current request allowing for concurrent issuing of requests. This allows for optimized bandwidth and reduced latency while maintaining appropriate command spacing to meet system memory protocol.

### **Command Overlap**

Command Overlap allows the insertion of the DRAM commands between the Activate, Pre-charge, and Read/Write commands normally used, as long as the inserted commands do not affect the currently executing command. Multiple commands can be issued in an overlapping manner, increasing the efficiency of system memory protocol.

### **Out-of-Order Scheduling**

While leveraging the Just-in-Time Scheduling and Command Overlap enhancements, the IMC continuously monitors pending requests to system memory for the best use of bandwidth and reduction of latency. If there are multiple requests to the same open page, these requests would be launched in a back to back manner to make optimum use of the open memory page. This ability to reorder requests on the fly allows the IMC to further reduce latency and increase bandwidth efficiency.

## **16.2.7 Data Scrambling**

The system memory controller incorporates a Data Scrambling feature to minimize the impact of excessive di/dt on the platform system memory VRs due to successive 1s and 0s on the data bus. Past experience has demonstrated that traffic on the data bus is not random and can have energy concentrated at specific spectral harmonics creating high di/dt which is generally limited by data patterns that excite resonance between the package inductance and on die capacitances. As a result, the system memory controller uses a data scrambling feature to create pseudo-random patterns on the system memory data bus to reduce the impact of any excessive di/dt.

## **16.2.8 Data Swapping**

By default, the processor supports on-board data swapping in two manners (for all segments and DRAM technologies):

- Bit swapping is allowed within each Byte for all DDR technologies.
- LPDDR5 - Byte swapping is allowed within each x16 sub channel.
- LPDDR5: Upper/Lower four x16 sub channels to be connected to x64 DRAM or two x32 DRAMs. Swapping between four upper to four lower x16 sub channels is not allowed.
- DDR4: Byte swapping is allowed within each x64 channel.
- DDR5: Byte swapping is allowed within a x16 word (BYTE 0-1 and BYTE 2-3) of the x32 Channel.

## **16.2.9 Ascending and Descending**

LPDDR5 support Ascending / descending that swap CA and CS signals connectivity order.

**Table 73. Ascending and Descending**

Ascending	Descending
CA6	CA0
CA5	CA1
CA4	CS_1
CA3	CS_0
CA2	CA2
CS_0	CA3
CS_1	CA4
CA1	CA5
CA0	CA6

### 16.2.10 DRAM Clock Generation

Each support rank has a differential clock pair for DDR4/5. Each sub-channel has a (CK\_P/N and WCK\_P/N) differential clock pair for LPDDR5.

### 16.2.11 DRAM Reference Voltage Generation

Read Vref is generated by the memory controller in all technologies. Write Vref is generated by the DRAM in all technologies. Command Vref is generated by the DRAM in LPDDR5 while the memory controller generates VrefCA per DIMM for DDR4. In all cases, it has small step sizes and is trained by MRC.

### 16.2.12 Data Swizzling

All Processor Lines does not have die-to-package DDR swizzling.

### 16.2.13 Error Correction With Standard RAM

In-Band error-correcting code (IBECC) correct single-bit memory errors in standard, non-ECC memory.

Supported only in Chrome systems.

## 16.3 Integrated Memory Controller (IMC) Power Management

The main memory is power managed during normal operation and in low-power ACPI C-states.

### 16.3.1 Disabling Unused System Memory Outputs

Any system memory (SM) interface signal that goes to a memory in which it is not connected to any actual memory devices (such as SODIMM connector is single-sided) is tri-stated. The benefits of disabling unused SM signals are:

- Reduced power consumption.



- Reduced possible overshoot/undershoot signal quality issues seen by the processor I/O buffer receivers caused by reflections from potentially unterminated transmission lines.

When a given rank is not populated, the corresponding control signals (CLK\_P/CLK\_N/CKE/ODT/CS) are not driven.

At reset, all rows should be assumed to be populated, until it can be proven that they are not populated. This is due to the fact that when CKE is tri-stated with a DRAMs present, the DRAMs are not ensured to maintain data integrity. CKE tri-state should be enabled by BIOS where appropriate, since at reset all rows should be assumed to be populated.

### 16.3.2 DRAM Power Management and Initialization

The processor implements extensive support for power management on the memory interface. The channel drives 2 CKE pins, one per rank.

The CKE is one of the power-saving means. When CKE is off, the internal DDR clock is disabled and the DDR power is reduced. The power-saving differs according to the selected mode and the DDR type used. For more information, refer to the IDD table in the DDR specification.

The processor supports four different types of power-down modes in package C0 state.

The different power-down modes supported are:

- **No power-down:** (CKE disable)
- **Active Power-down (APD):** This mode is entered if there are open pages when de-asserting CKE. In this mode the open pages are retained. Power-saving in this mode is the lowest. Power consumption of DDR is defined by IDD3P. Exiting this mode is fined by tXP – a small number of cycles.
- **Pre-charged Power-down (PPD):** This mode is entered if all banks in DDR are pre-charged when de-asserting CKE. Power-saving in this mode is intermediate – better than APD. Power consumption is defined by IDD2P. Exiting this mode is defined by tXP. The difference from APD mode is that when waking-up, all page-buffers are empty.)

The CKE is determined per rank, whenever it is inactive. Each rank has an idle counter. The idle-counter starts counting as soon as the rank has no accesses, and if it expires, the rank may enter power-down while no new transactions to the rank arrive to queues. The idle-counter begins counting at the last incoming transaction arrival. It is important to understand that since the power-down decision is per rank, the IMC can find many opportunities to power down ranks, even while running memory intensive applications; the savings are significant (may be few Watts, according to DDR specification). This is significant when the channel is populated with more ranks.

Selection of power modes should be according to power-performance or a thermal trade-off of a given system:

- When trying to achieve maximum performance and power or thermal consideration is not an issue: use no power-down
- In a system which tries to minimize power-consumption, try using the deepest power-down mode possible

- In high-performance systems with dense packaging (that is, tricky thermal design) the power-down mode should be considered in order to reduce the heating and avoid DDR throttling caused by the heating.

The idle timer expiration count defines the # of DCLKs that a rank is idle that causes entry to the selected power mode. As this timer is set to a shorter time the IMC will have more opportunities to put the DDR in power-down. There is no BIOS hook to set this register. Customers choosing to change the value of this register can do it by changing it in the BIOS. For experiments, this register can be modified in real time if BIOS does not lock the IMC registers.

#### 16.3.2.1 Initialization Role of CKE

During power-up, CKE is the only input to the SDRAM that has its level recognized (other than the reset pin) once power is applied. It should be driven LOW by the DDR controller to make sure the SDRAM components float DQ and DQS during power-up. CKE signals remain LOW (while any reset is active) until the BIOS writes to a configuration register. Using this method, CKE is ensured to remain inactive for much longer than the specified 200 micro-seconds after power and clocks to SDRAM devices are stable. In LPDDR5/DDR5, there is no CKE pin and the power management roll is assumed by the CS signals.

#### 16.3.2.2 Conditional Self-Refresh

During S0 idle state, system memory may be conditionally placed into self-refresh state when the processor is in package C3 or deeper power state. Refer to [Intel® Rapid Memory Power Management \(Intel® RMPM\)](#) on page 121 for more details on conditional self-refresh with Intel HD Graphics enabled.

When entering the S3 – Suspend-to-RAM (STR) state or S0 conditional self-refresh, the processor IA core flushes pending cycles and then enters SDRAM ranks that are not used by the processor graphics into self-refresh. The CKE signals remain LOW so the SDRAM devices perform self-refresh.

The target behavior is to enter self-refresh for package C3 or deeper power states as long as there are no memory requests to service.

#### 16.3.2.3 Dynamic Power-Down

Dynamic power-down of memory is employed during normal operation. Based on idle conditions, a given memory rank may be powered down. The IMC implements aggressive CKE control to dynamically put the DRAM devices in a power-down state.

The processor IA core controller can be configured to put the devices in active power down (CKE de-assertion with open pages) or pre-charge power-down (CKE de-assertion with all pages closed). Pre-charge power-down provides greater power savings but has a bigger performance impact, since all pages will first be closed before putting the devices in power-down mode.

If dynamic power-down is enabled, all ranks are powered up before doing a refresh cycle and all ranks are powered down at the end of the refresh.

#### 16.3.2.4 DRAM I/O Power Management

Unused signals should be disabled to save power and reduce electromagnetic interference. This includes all signals associated with an unused memory channel. Clocks, CKE, ODT, and CS signals are controlled per DIMM rank and will be powered down for unused ranks.

The I/O buffer for an unused signal should be tri-stated (output driver disabled), the input receiver (differential sense-amp) should be disabled. The input path should be gated to prevent spurious results due to noise on the unused signals (typically handled automatically when input receiver is disabled).

#### 16.3.3 DDR Electrical Power Gating

The DDR I/O of the processor supports Electrical Power Gating (DDR-EPG) while the processor is at C3 or deeper power state.

In C3 or deeper power state, the processor internally gates VDDQ and VDD2 for the majority of the logic to reduce idle power while keeping all critical DDR pins such as CKE and VREF in the appropriate state.

In C7 or deeper power state, the processor internally gates VCCSA for all non-critical state to reduce idle power.

In S3 or C-state transitions, the DDR does not go through training mode and will restore the previous training information.

#### 16.3.4 Power Training

BIOS MRC performing Power Training steps to reduce DDR I/O power while keeping reasonable operational margins still guaranteeing platform operation. The algorithms attempt to weaken ODT, driver strength and the related buffers parameters both on the MC and the DRAM side and find the best possible trade-off between the total I/O power and the operating margins using advanced mathematical models.

## 17.0 USB-C\* Sub System (TCSS)

USB-C\* is a cable and connector specification defined by USB-IF.

The USB-C sub-system supports USB3, DPoC (DisplayPort over Type-C) protocols. The USB-C sub-system can also be configured as native DisplayPort or HDMI interfaces, for more information refer to [Display](#) on page 197 .

USB-C solution brand requires USB2, USB3 (10 Gbps), USB3/DP implemented at the connector.

### 17.1 General Capabilities

- xHCI (USB 3 host controller) and xDCI (USB 3 device controller) implemented in the processor in addition to the controllers in the PCH.
- No support for USB Type-A on the processor side
- Support power saving when USB-C\* disconnected.
- Support up to two simultaneous ports.
- DbC Enhancement for Low Power Debug until Pkg C6
- Host
  - Wake capable on each host port from S0i3, Device Wake.
- Device
  - Aggregate BW through xHCI controller of at least 3 GB/s
  - Wake capable on host initiated wakes when the system is in S0i3, Sx Available on all ports
- Port Routing Control for Dual Role Capability
  - Needs to support SW/FW and ID pin based control to detect host versus device attach
  - SW mode requires PD controller or other FW to control

**Table 74. USB-C\* Port Configuration**

	Port	Supported Features
Group A	TCP 0	USB 3 <sup>3</sup> DisplayPort <sup>1</sup> HDMI <sup>2</sup>
	TCP 1	
<p><i>Notes:</i> 1. Supported on Type-C or Native connector            2. Supported only on Native connector.            3. USB 3 supported link rates:                a. USB 3 Gen 1x1 (5 Gbps)                b. USB 3 Gen 2x1 (10 Gbps)            4. USB 2 interface supported over Type-C connector, sourced from PCH.            5. USB Type-A connector is not supported.            6. Display interface can be connected directly to a DP/HDMI/Type-C port.</p>		

**Table 75. USB-C\* Lanes Configuration**

Lane1	Lane2	Comments
USB3	No connect	Any combination of <ul style="list-style-type: none"> <li>• USB3.2 Gen 1x1 (5 Gbps)</li> <li>• USB3.2 Gen 2x1 (10 Gbps)</li> </ul>
No connect	USB3	
USB3	DPx2	Any of HBR3/HBR2/HBR1/RBR for DP and USB3.2 (10 Gbps)
DPx2	USB3	
DPx4		Both lanes at the same DP rate - no support for 2x DPx2 USB-C connector

## 17.2 USB-C Sub-system xHCI/xDCI Controllers

The processor supports xHCI/xDCI controllers. The native USB 3 path proceeds from the memory directly to PHY.

### 17.2.1 USB 3 Controllers

#### 17.2.1.1 Extensible Host Controller Interface (xHCI)

Extensible Host Controller Interface (xHCI) is an interface specification that defines Host Controller for a universal Serial Bus (USB 3), which is capable of interfacing with USB 1.x, 2.0, and 3.x compatible devices.

In case that a device (example, USB3 mouse) was connected to the computer, the computer will work as Host and the xHCI will be activated inside the CPU.

The xHCI controller support link rate of up to USB 3.2 Gen 2x1 (10G).

#### 17.2.1.2 Extensible Device Controller Interface (xDCI)

Extensible Device Controller Interface (xDCI) is an interface specification that defines Device Controller for a universal Serial Bus (USB 3), which is capable of interfacing with USB 1.x, 2.0, and 3.x compatible devices.

In case that the computer is connected as a device (example, tablet connected to desktop) to another computer then the xDCI controller will be activated inside the device and will talk to the Host at the other computer.

The xDCI controller support link rate of up to USB 3.2 Gen 1x1 (5G).

---

#### NOTE

These controllers are instantiated in the processor die as a separate PCI function functionality for the USB-C\* capable ports.

---

### Type-C Port (TCP) and USB2.0 Port Pairing

USB2 port number 1, 2, 5, 6 can be paired with TCP ports 0 or 1.

Port#	USB2 Port Number that can be Paired with TCP Ports
TCP Port # 0	1, 2, 5, 6
TCP Port # 1	1, 2, 5, 6

### 17.3 USB-C Sub-System Display Interface

Refer [Display](#) on page 197.

### 17.4 USB Type-C Signals

Signal Name	Description	Dir.	Link Type
TCP[1:0]_TX_P[1:0] TCP[1:0]_TX_N[1:0]	TX Data Lane.	O	Diff
TCP[1:0]_TXRX_P[1:0] TCP[1:0]_TXRX_N[1:0]	RX Data Lane, also serves as the secondary TX data lane.	I/O	Diff
TCP[1:0]_AUX_P TCP[1:0]_AUX_N	Common Lane AUX-PAD.	I/O	Diff
TCPO_RCOMP	Type-C Resistance Compensation.	N/A	Diff

## 18.0 Universal Serial Bus (USB)

The PCH implements an xHCI USB 3.2 controller which provides support for up to 8 USB 2.0 signal pairs and 4 USB 3.2 signal pairs. The xHCI controller supports wake up from sleep states S1-S4. The xHCI controller supports up to 64 devices for a maximum number of 2048 Asynchronous endpoints (Control / Bulk) or maximum number of 128 Periodic endpoints (Interrupt / isochronous).

Each walk-up USB 3.2 capable port must include USB 3.2 and USB 2.0 signaling.

**Table 76. Acronyms**

Acronyms	Description
xHCI	eXtensible Host Controller Interface

**Table 77. References**

Specification	Location
USB 3.2 Specification	<a href="http://www.usb.org">www.usb.org</a>
USB 2.0 Specification	

### 18.1 Functional Description

This section provides information on the following topics:

1. eXtensible Host Controller Interface (xHCI) Controller
2. USB Dual Role Support - eXtensible Device Controller Interface (xDCI) Controller

#### 18.1.1 eXtensible Host Controller Interface (xHCI) Controller

The eXtensible Host Controller Interface (xHCI) allows data transfer speed up to 10 Gb/s for USB 3.2 Gen 2x1 ports, and 5 Gb/s for USB 3.2 Gen 1x1 ports. The xHCI supports SuperSpeed USB 10 Gbps, SuperSpeed USB 5 Gbps, High-Speed (HS), Full-Speed (FS) and Low-Speed (LS) traffic on the bus. The xHCI supports USB Debug port on all the USB ports. The xHCI also supports USB Attached SCSI Protocol (UASP).

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

The USB subsystem also supports Dual Role Capability. The xHCI is paired with a standalone eXtensible Device Controller Interface (xDCI) to provide dual role functionality. The USB subsystem incorporates a xDCI USB 3.2 Gen 1x1 (5 Gb/s) device controller. The dual role capability splits the support for SuperSpeed USB 5 Gbps on the CPU xDCI controller, and High-Speed (HS) on the PCH xDCI controller. These device controllers are instantiated as a separate PCI function. The USB implementation is compliant to the Device specification and supports host/device only through the integrated USB Type-C\* connector.

The xDCI shares all USB ports with the host controller, with the ownership of the port being decided based the USB Power Delivery specification. Since all the ports support device mode, xDCI enabling must be extended by System BIOS and EC. While the port is mapped to the device controller, the host controller Rx detection must always indicate a disconnected port. Only one port can be connected (and active) to the device controller at one time. Any subsequently connection will not be established.

### 18.1.3 **AUX BIAS Control - USB Type-C Implementation with no Retimer**

On processor, which support integrated USB Type-C subsystem, the AUX BIAS control is required on the USB Type-C implementation (without retimer) for orientation connections.

The functionality is muxed with certain GPIO pins. Refer to the GPIO implementation document for more information on the muxing and supported GPIO pin on the specific platform. In order to use the GPIO pin correctly for AUX BIAS control, the correct native functionality need to be configured and the correct Virtual Wire Index bit position need to be programmed in the BIOS policy.



**Figure 16. GPIO - Virtual Wire Index Bit Mapping**

GPIO Pin Group	Virtual Wire Index	Bit Position*
USB_C_GPP_[A7:A0]	10h	[7h:0h]
USB_C_GPP_[A15:A8]	11h	[7h:0h]
USB_C_GPP_[A23:A16]	12h	[7h:0h]
USB_C_GPP_[B7:B0]	13h	[7h:0h]
USB_C_GPP_[B15:B11, B8]	14h	[7h:0h]
USB_C_GPP_[B23, B18:B16]	15h	[7h:0h]
USB_C_GPP_[C7:C0]	13h	[7h:0h]
USB_C_GPP_[D7:D0]	10h	[7h:0h]
USB_C_GPP_[D15:D8]	11h	[7h:0h]
USB_C_GPP_[D19:D16]	12h	[3h:0h]
USB_C_GPP_[E7:E0]	16h	[7h:0h]
USB_C_GPP_[E15:E8]	17h	[7h:0h]
USB_C_GPP_[E23:E16]	18h	[7h:0h]
USB_C_GPP_[F7:F0]	10h	[7h:0h]
USB_C_GPP_[F15:F10]	11h	[7h:0h]
USB_C_GPP_[F18:F16, F23:F22]	12h	[7h:0h]
USB_C_GPP_[H3:H0]	12h	[7h:4h]
USB_C_GPP_[H11:H4]	13h	[7h:0h]
USB_C_GPP_[H13:H12, H15, H19:H17]	14h	[7h:0h]
USB_C_GPP_[H23:H20]	15h	[3h:0h]
USB_C_GPP_[I11:I7, I5]	16h	[7h:0h]
USB_C_GPP_[I18:I12]	17h	[7h:0h]

**NOTE**

1. The bit position corresponds to each corresponding GPIO pin in the group.  
Example: the bit position for USB\_C\_GPP\_A0 is bit 0h in Virtual Wire Index 10h.

## 18.2 Signal Description

Name	Type	Description
PCIE1_RXN / <b>USB32_1_RXN</b> PCIE1_RXP / <b>USB32_1_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 1:</b> These are USB 3.2-based high-speed differential signals for Port 1. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals.

*continued...*

Name	Type	Description
		<i>Note:</i> Use FITC to set the soft straps that select this port as PCIe* Port 1.
PCIE1_TXN / <b>USB32_1_TXN</b> PCIE1_TXP / <b>USB32_1_TXP</b>	O	<b>USB 3.2 Differential Transmit Pair 1:</b> These are USB 3.2-based high-speed differential signals for Port 1. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals. <i>Note:</i> Use FITC to set the soft straps that select this port as PCIe* Port 1.
PCIE2_RXN / <b>USB32_2_RXN</b> PCIE2_RXP / <b>USB32_2_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 2:</b> These are USB 3.2-based high-speed differential signals for Port 2. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals. <i>Note:</i> Use FITC to set the soft straps that select this port as PCIe* Port 2.
PCIE2_TXN / <b>USB32_2_TXN</b> PCIE2_TXP / <b>USB32_2_TXP</b>	O	<b>USB 3.2 Differential Transmit Pair 2:</b> These are USB 3.2-based high-speed differential signals for Port 2. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals. <i>Note:</i> Use FITC to set the soft straps that select this port as PCIe* Port 2.
PCIE3_RXN / <b>USB32_3_RXN</b> PCIE3_RXP / <b>USB32_3_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 3:</b> These are USB 3.2-based high-speed differential signals for Port 3. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals. <i>Note:</i> Use FITC to set the soft straps that select this port as PCIe* Port 3.
PCIE3_TXN / <b>USB32_3_TXN</b> PCIE3_TXP / <b>USB32_3_TXP</b>	O	<b>USB 3.2 Differential Transmit Pair 3:</b> These are USB 3.2-based high-speed differential signals for Port 3. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals. <i>Note:</i> Use FITC to set the soft straps that select this port as PCIe* Port 3.
PCIE4_RXN / <b>USB32_4_RXN</b> PCIE4_RXP / <b>USB32_4_RXP</b>	I	<b>USB 3.2 Differential Receive Pair 4:</b> These are USB 3.2-based high-speed differential signals for Port 4. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals. <i>Note:</i> Use FITC to set the soft straps that select this port as PCIe* Port 4.
PCIE4_TXN / <b>USB32_4_TXN</b> PCIE4_TXP / <b>USB32_4_TXP</b>	O	<b>USB 3.2 Differential Transmit Pair 4:</b> These are USB 3.2-based high-speed differential signals for Port 4. The signal should be mapped to a USB connector with one of the OC (overcurrent) signals. <i>Note:</i> Use FITC to set the soft straps that select this port as PCIe* Port 4.
<b>USB2P_1</b> <b>USB2N_1</b>	I/O	<b>USB 2.0 Port 1 Transmit/Receive Differential Pair 1:</b> This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
<b>USB2P_2</b> <b>USB2N_2</b>	I/O	<b>USB 2.0 Port 2 Transmit/Receive Differential Pair 2:</b> This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
<b>USB2P_3</b> <b>USB2N_3</b>	I/O	<b>USB 2.0 Port 3 Transmit/Receive Differential Pair 3:</b> This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
<b>continued...</b>		



Name	Type	Description
<b>USB2P_4</b> <b>USB2N_4</b>	I/O	<b>USB 2.0 Port 4 Transmit/Receive Differential Pair 4:</b> This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
<b>USB2P_5</b> <b>USB2N_5</b>	I/O	<b>USB 2.0 Port 5 Transmit/Receive Differential Pair 5:</b> This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
<b>USB2P_6</b> <b>USB2N_6</b>	I/O	<b>USB 2.0 Port 6 Transmit/Receive Differential Pair 6:</b> This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
<b>USB2P_7</b> <b>USB2N_7</b>	I/O	<b>USB 2.0 Port 7 Transmit/Receive Differential Pair 7:</b> This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
<b>USB2P_8</b> <b>USB2N_8</b>	I/O	<b>USB 2.0 Port 8 Transmit/Receive Differential Pair 8:</b> This USB 2.0 signal pair are routed to xHCI controller and should be mapped to a USB connector with one of the OC (overcurrent) signals.
GPP_E9 / <b>USB_OC0#</b> / ISH_GP4	I	<p><b>Overcurrent Indicators:</b> This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred.</p> <p>When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required.</p> <p><i>Notes:</i> 1. OC# pins are not 5V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..</p>
GPP_A14 / <b>USB_OC1#</b> / DDSP_HP3 / DISP_MISC3	I	<p><b>Overcurrent Indicators:</b> This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred.</p> <p>When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required.</p> <p><i>Notes:</i> 1. OC# pins are not 5V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..</p>
GPP_A15 / <b>USB_OC2#</b> / DDSP_HP4 / DISP_MISC4	I	<p><b>Overcurrent Indicators:</b> This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred.</p> <p>When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required.</p> <p><i>Notes:</i> 1. OC# pins are not 5V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..</p>
GPP_A16 / <b>USB_OC3#</b> / ISH_GP5	I	<p><b>Overcurrent Indicators:</b> This signal set the corresponding bit in the xHCI controller to indicate that an overcurrent condition has occurred.</p> <p>When configured as OC# pin, a 10 kΩ pull-up resistor is required to be connected to the power-rail. When this pin is configured as GPIO, no pull-up resistor is required.</p> <p><i>Notes:</i> 1. OC# pins are not 5V tolerant. 2. OC# pins can be shared between USB ports. 3. Each USB connector should only have one OC# pin protection..</p>
<i>continued...</i>		

Name	Type	Description
<b>USB_ID</b>	I	ID detect for USB Device mode. Note: This HW signal is not used on the PCH for dual role mode selection. The switching of USB port role is done through message from EC/PD over SMLink1. This signal should be connected to ground.
<b>USB2_COMP</b>	I	USB Resistor Bias, analog connection points for an external resistor $113\ \Omega \pm 1\%$ connected to GND.
<b>USB_VBUSSENSE</b>	I	VBUS Sense for USB Device mode. This HW signal is not used on the PCH for USB device mode functionality. This signal should be connected to ground.

**NOTE**

For TCP and USB2.0 port pairing, refer to [Type-C Port \(TCP\) and USB2.0 Port Pairing](#) on page 165.

### 18.3 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
<b>USB2P_[8:1]</b>	Internal Pull-down	14.25–24.8 k $\Omega$	1
<b>USB2N_[8:1]</b>	Internal Pull-down	14.25–24.8 k $\Omega$	1
<b>USB_ID</b>	Internal Weak Pull-up	14.25–24.8 k $\Omega$	If this signal is not in use, then the pin shall be connected directly to ground.

*Note:* 1. Series resistors (45 Ohm  $\pm 10\%$ )

### 18.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>2</sup>	Immediately After Reset <sup>2</sup>	S4/S5	Deep Sx
<b>USB32_[4:1]_RXN</b> <b>USB32_[4:1]_RXP</b>	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
<b>USB32_[4:1]_TXN</b> <b>USB32_[4:1]_TXP</b>	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
<b>USB2N_[8:1]</b>	DSW	Internal Pull-down	Internal Pull-down	Internal Pull-down	Internal Pull-down
<b>USB2P_[8:1]</b>	DSW	Internal Pull-down	Internal Pull-down	Internal Pull-down	Internal Pull-down
<b>USB_OC0#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>USB_OC1#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>USB_OC2#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>USB_OC3#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>USB_VBUSSENSE</b>	Primary	Undriven	Undriven	Undriven	OFF

**continued...**



Signal Name	Power Plane	During Reset <sup>2</sup>	Immediately After Reset <sup>2</sup>	S4/S5	Deep Sx
<b>USB_ID<sup>1</sup></b>	Primary	Internal Pull-up	Undriven/Internal Pull-up	Undriven/Internal Pull-up	OFF
<b>USB2_COMP</b>	Primary	Undriven	Undriven	Undriven	OFF
Notes: 1. The USB_ID pin is pulled-up internally. 2. Reset reference for primary well pins is RSMRST# and DSW well pins is DSW_PWROK.					

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

**Table 78. Acronym**

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

### 19.1 Functional Description

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

#### 19.1.1 Interrupt Generation

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

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

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

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

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

## 19.1.2 PCI Express\* Power Management

### S3/S4/S5 Support

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

Prior to entering S3, software is required to put each device into D3HOT. When a device is put into D3HOT, it will initiate entry into a L1 link state by sending a PM\_Enter\_L1 DLLP. Under normal operating conditions when the root ports sends the PME\_Turn\_Off message, the link will be in state L1. However, when the root port is instructed to send the PME\_Turn\_Off message, it will send it whether or not the link was in L1. Endpoints attached to the PCH can make no assumptions about the state of the link prior to receiving a PME\_Turn\_Off message.

### Device Initiated PM\_PME Message

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

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

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

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

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

### SMI/SCI Generation

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

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

### Latency Tolerance Reporting (LTR)

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

---

#### NOTE

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

---

### 19.1.3 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.

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

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

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

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



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

### 19.1.5 Advanced Error Reporting

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

### 19.1.6 Single - Root I/O Virtualization (SR - IOV)

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

#### Alternative Routing - ID Interpretation (ARI)

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

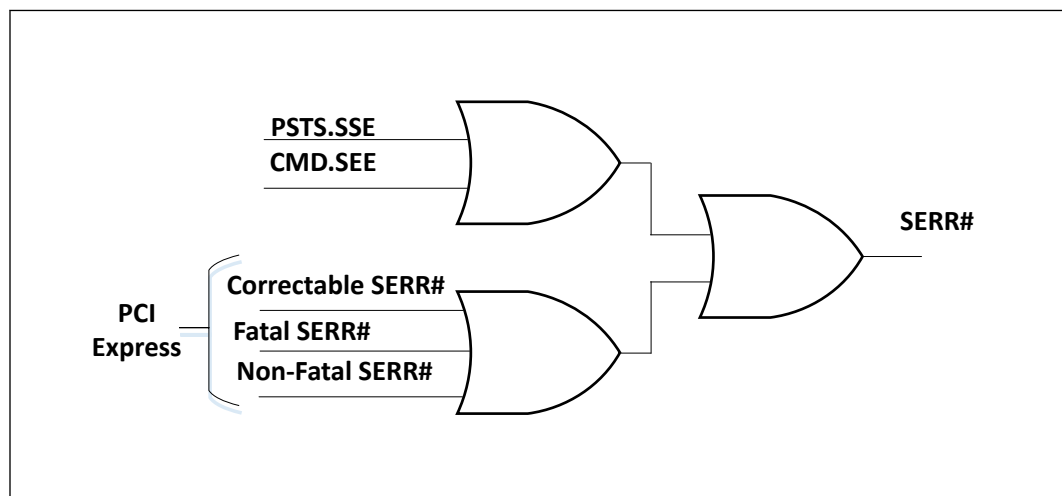
#### Access Control Services (ACS)

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

### 19.1.7 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 17. Generation of SERR# to Platform



### 19.1.8 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

#### Presence Detection

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

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

#### SMI/SCI Generation

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

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

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

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

### 19.1.9 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.

---

#### NOTE

The polarity inversion does not imply direction inversion or direction reversal; that is, the Tx differential pair from one device must still connect to the Rx differential pair on the receiving device, per the PCIe\* Base Specification. Polarity Inversion is not the same as "PCI Express\* Controller Lane Reversal".

---

### 19.1.10 Precision Time Measurement (PTM)

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

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

## 19.2 Signal Description

Name	Type	Description
<b>PCIE1_TXP</b> / USB32_1_TXP <b>PCIE1_TXN</b> / USB32_1_TXN <b>PCIE2_TXP</b> / USB32_2_TXP <b>PCIE2_TXN</b> / USB32_2_TXN <b>PCIE3_TXP</b> / USB32_3_TXP <b>PCIE3_TXN</b> / USB32_3_TXN <b>PCIE4_TXP</b> / USB32_4_TXP <b>PCIE4_TXN</b> / USB32_4_TXN <b>PCIE7_TXP</b> <b>PCIE7_TXN</b> <b>PCIE9_TXP</b> / UFS10_TXP <b>PCIE9_TXN</b> / UFS10_TXN <b>PCIE10_TXP</b> / UFS11_TXP <b>PCIE10_TXN</b> / UFS11_TXN <b>PCIE11_TXP</b> / SATA0_TXP <b>PCIE11_TXN</b> / SATA0_TXN <b>PCIE12_TXP</b> / SATA1_TXP <b>PCIE12_TXN</b> / SATA1_TXN	O	<b>PCI Express* Differential Transmit Pairs</b> These are PCI Express* based outbound high-speed differential signals
<b>PCIE1_RXP</b> / USB32_1_RXP <b>PCIE1_RXN</b> / USB32_1_RXN <b>PCIE2_RXP</b> / USB32_2_RXP <b>PCIE2_RXN</b> / USB32_2_RXN <b>PCIE3_RXP</b> / USB32_3_RXP <b>PCIE3_RXN</b> / USB32_3_RXN <b>PCIE4_RXP</b> / USB32_4_RXP <b>PCIE4_RXN</b> / USB32_4_RXN <b>PCIE7_RXP</b> <b>PCIE7_RXN</b> <b>PCIE9_RXP</b> / UFS10_RXP <b>PCIE9_RXN</b> / UFS10_RXN <b>PCIE10_RXP</b> / UFS11_RXP <b>PCIE10_RXN</b> / UFS11_RXN <b>PCIE11_RXP</b> / SATA0_RXP <b>PCIE11_RXN</b> / SATA0_RXN <b>PCIE12_RXP</b> / SATA1_RXP <b>PCIE12_RXN</b> / SATA1_RXN	I	<b>PCI Express* Differential Receive Pairs</b> These are PCI Express* based inbound high-speed differential signals
GPP_H15/DDPB_CTRLCLK/ <b>PCIE_LINK_DOWN</b>	O	<b>PCIE_LINK_DOWN Output</b> PCIe link failure debug signal. PCH PCIe Root Port(s) will assert this signal when a link down event occurs and is detected. For example when a link fails to train during an L1 sub-state exit event.

### 19.3 I/O Signal Planes and States

**Table 80. Power Plane and States for PCI Express\* Signals**

Signal Name	Type	Power Plane	During Reset <sup>2</sup>	Immediately After Reset <sup>2</sup>	S3/S4/S5	Deep Sx
PCIE_TXP/ PCIE_TXN	O	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
PCIE_RXP/ PCIE_RXN	I	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF

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

### 19.4 PCI Express\* Port Support Feature Details

**Table 81. PCI Express\* Port Feature Details**

Max Transfer Rate	Max Device (Ports)	Max Lanes	PCIe* Gen Type	Encoding	Transfer Rate (MT/s)	Theoretical Max Bandwidth (GB/s)		
						x1	x2	x4
8 GT/s (Gen3)	5	9	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

*Note:* 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

**Figure 18. Supported PCI Express\* Link Configurations**

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

---

**NOTES**

1. If PCIe controller #3 is used for UFS than Lane reversal is not supported.
  2. If all the four ports of PCIe controller #3 are used for PCIe, than Lane reversal is supported.
  3. Enable UFS will turn both lanes (8,9) become UFS regardless, and cannot be used for PCIe.
  4. This platform supports one internal storage device. Dual/Concurrent internal storage is not validated by Intel.
  5. RP# refers to a specific PCH PCI Express\* Root Port #; for example RP3 = PCH PCI Express\* Root Port 3
  6. A PCIe\* Lane is composed of a single pair of Transmit (TX) and Receive (RX) differential pairs, for a total of four data wires per PCIe\* Lane (such as, PCIe[3]\_TXP/ PCIe[3]\_TXN and PCIe[3]\_RXP/ PCIe[3]\_RXN make up PCIe Lane 3). A connection between two PCIe\* devices is known as a PCIe\* Link, and is built up from a collection of one or more PCIe\* Lanes which make up the width of the link (such as bundling 2 PCIe\* Lanes together would make a x2 PCIe\* Link). A PCIe\* Link is addressed by the lowest number PCIe\* Lane it connects to and is known as the PCIe\* Root Port (such as a x2 PCIe\* Link connected to PCIe\* Lanes 3 and 4 would be called x2 PCIe\* Root Port 3).
  7. The PCIe\* Lanes can be configured independently from one another but the max number of configured Root Ports (Devices) must not be exceeded
    - A maximum of 5 PCIe\* Root Ports (or devices) can be enabled
  8. Unidentified lanes within a PCIe\* Link Configuration are disabled but their physical lanes are used for the identified Root Port
  9. The PCH PCIe\* Root Ports can be configured to map to any of the SRCCLKREQ# PCIe\* clock request signals and the CLKOUT\_PCIE\_P/N PCIe\* differential clock signal pairs.
  10. Lane Reversal Supported Motherboard PCIe\* Configurations = 1x4, 2x1+1x2, and 2x2
    - The 2x1+1x2 configuration is enabled by setting the PCIe\* Controller soft straps to 1x2+2x1 with Lane Reversal Enabled
    - 1x4 = 1x4 with Lane Reversal Disabled, 1x4 LR = 1x4 with Lane Reversal Enabled
    - 2x2 = 2x2 with Lane Reversal Disabled, 2x2 LR = 2x2 with Lane Reversal Enabled
  11. For unused SATA/PCIe\* and USB 3.2/PCIe\* Combo Lanes, the unused lanes must be statically assigned to PCIe\*, SATA, or USB 3.2 via the SATA/PCIe\* and USB 3.2/PCIe\* Combo Port Soft Straps through the Intel® Flash Image Tool (Intel® FIT) tool.
-

## 20.0 Serial ATA (SATA)

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

**Table 82. Acronyms**

Acronyms	Description
AHCI	Advanced Host Controller Interface
DMA	Direct Memory Access
DEVSLP	Device Sleep
IDE	Integrated Drive Electronics
SATA	Serial Advanced Technology Attachment

**Table 83. References**

Specification	Location
Serial ATA Specification, Revision 3.2	<a href="https://www.sata-io.org">https://www.sata-io.org</a>
Serial ATA II: Extensions to Serial ATA 1.0, Revision 1.0	<a href="https://www.sata-io.org">https://www.sata-io.org</a>
Serial ATA II Cables and Connectors Volume 2 Gold	<a href="https://www.sata-io.org">https://www.sata-io.org</a>
Advanced Host Controller Interface Specification	<a href="http://www.intel.com/content/www/us/en/io/serial-ata/ahci.html">http://www.intel.com/content/www/us/en/io/serial-ata/ahci.html</a>

### 20.1 Functional Description

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

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

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

#### 20.1.1 SATA 6 Gb/s Support

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

## 20.1.2 SATA Feature Support

The SATA controller is capable of supporting all AHCI 1.3 and AHCI 1.3.1, refer to the Intel web site on Advanced Host Controller Interface Specification for current specification status: <http://www.intel.com/content/www/us/en/io/serial-ata/ahci.html>.

For capability details, refer to SATA controller register (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)

## 20.1.3 Hot - Plug Operation

The PCH SATA controller supports Hot- Plug Surprise removal and Insertion Notification. An internal SATA port with a Mechanical Presence Switch can support PARTIAL and SLUMBER with Hot - Plug Enabled. Software can take advantage of the power savings in the low power states while enabling Hot - Plug operation. Refer to Chapter 7 of the AHCI specification for details.

## 20.1.4 Power Management Operation

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

### Power State Mappings

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

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

- **D0** – Device is working and instantly available.
- **D1** – Device enters when it receives a STANDBY IMMEDIATE command. Exit latency from this state is in seconds.
- **D3** – From the SATA device’s perspective, no different than a D1 state, in that it is entered using the STANDBY IMMEDIATE command. However, an ACPI method is also called which will reset the device and then cut its power.

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

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

- **PHY READY** – PHY logic and PLL are both on and in active state.
- **Partial** – PHY logic is powered up, and in a reduced power state. The link PM exit latency to active state maximum is 10 ns.

- **Slumber** – PHY logic is powered up, and in a reduced power state. The link PM exit latency to active state maximum is 10 ms.
- **Devslp** – PHY logic is powered down. The link PM exit latency from this state to active state maximum is 20 ms, unless otherwise specified by DETO in Identify Device Data Log page 08h (Refer to SATA Rev3.2 Gold specification).

Since these states have much lower exit latency than the ACPI D1 and D3 states, the SATA controller specification defines these states as sub-states of the device D0 state.

### Power State Transitions

- **Partial and Slumber State Entry/Exit**

The partial and slumber states save interface power when the interface is idle. It would be most analogous to CLKRUN# (in power savings, not in mechanism), where the interface can have power saved while no commands are pending. The SATA controller defines PHY layer power management (as performed using primitives) as a driver operation from the host side, and a device proprietary mechanism on the device side. The SATA controller accepts device transition types, but does not issue any transitions as a host. All received requests from a SATA device will be ACKed.

When an operation is performed to the SATA controller such that it needs to use the SATA cable, the controller must check whether the link is in the Partial or Slumber states, and if so, must issue a COMWAKE to bring the link back online. Similarly, the SATA device must perform the same COMWAKE action.

---

#### NOTE

SATA devices shall not attempt to wake the link using COMWAKE/COMINIT when no commands are outstanding and the interface is in Slumber.

---

- **Devslp State Entry/Exit**

Device Sleep (DEVSLP) is a host - controlled SATA interface power state. To support a hardware autonomous approach that is software agnostic Intel is recommending that BIOS configure the AHCI controller and the device to enable Device Sleep. This allows the AHCI controller and associated device to automatically enter and exit Device Sleep without the involvement of OS software.

To enter Device Sleep the link must first be in Slumber. By enabling HIPM (with Slumber) or DIPM on a Slumber capable device, the device/host link may enter the DevSleep Interface Power state.

The device must be DevSleep capable. Device Sleep is only entered when the link is in slumber, therefore when exiting the Device Sleep state, the device must resume with the COMWAKE out - of - band signal (and not the COMINIT out - of - band signal). Assuming Device Sleep was asserted when the link was in slumber, the device is expected to exit DEVSLP to the DR\_Slumber state. Devices that do not support this feature will not be able to take advantage of the hardware automated entry to Device Sleep that is part of the AHCI 1.3.1 specification and supported by Intel platforms.

- **Device D1 and D3 States**



These states are entered after some period of time when software has determined that no commands will be sent to this device for some time. The mechanism for putting a device in these states does not involve any work on the host controller, other than sending commands over the interface to the device. The command is most likely to be used in ATA/ATAPI is the “STANDBY IMMEDIATE” command.

- **Host Controller D3<sub>HOT</sub> State**

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

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

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

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

#### Low Power Platform Consideration

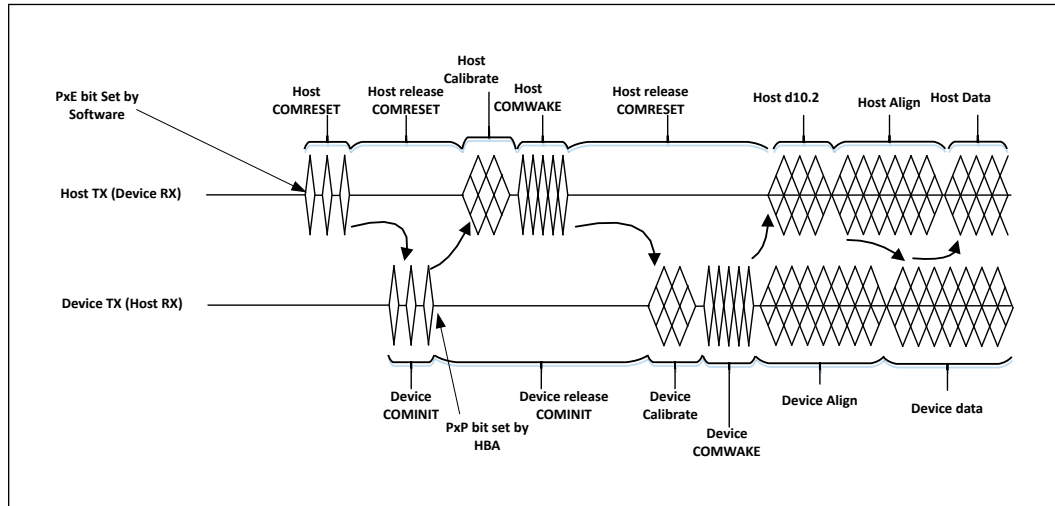
When low power feature is enabled, the Intel SATA controller may power off PLLs or OOB detection circuitry while in the Slumber link power state. As a result, a device initiated wake may not be recognized by the host. For example, when the low power feature is enabled it can prevent a Zero Power ODD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

The SATA MPHY Dynamic Power Gating (PHYDPGEPx) can be enabled/disabled for each SATA ports.

### 20.1.5 SATA Device Presence

The flow used to indicate SATA device presence is shown in the Figure below. The ‘PxE’ bit refers to PCS.P[1:0]E bits, depending on the port being checked and the ‘PxP’ bits refer to the PCS.P[1:0]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 19. Flow for Port Enable/Device Present Bits



### 20.1.6 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.

**NOTE**

SATA\_LED# signal is muxed with SPKR signal. Only one function can be used at a time.

### 20.1.7 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 initiator—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 initiator/target designation for SATA devices—each device is treated as a initiator—and hardware assisted native command queuing. AHCI also provides usability enhancements such as hot - plug and advanced power management. AHCI requires appropriate software support (such as, an AHCI driver) and for some features, hardware support in the SATA device or additional platform hardware. Visit the Intel web site for current information on the AHCI specification.

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

---

**NOTE**

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

---

## 20.1.8 Enclosure Management (SGPIO Signals)

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

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

---

**NOTE**

These signals are not related to SATALED#.

---

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

---

**NOTE**

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

---

### Mechanism

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

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

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

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

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

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

---

**NOTE**

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

---

**Message Format**

Messages shall be constructed with a one DWord header that describes the message to be sent followed by the actual message contents. The first DWord shall be constructed as shown in Enclosure Management Message Format (EM\_MF) register, refer to Intel® Processor and Intel® Core™ i3 N-series Datasheet Volume 2 of 2 (#759604).

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 Intel® Processor and Intel® Core™ i3 N-series Datasheet Volume 2 of 2 (#759604). It is the responsibility of software to ensure the content of the message format is correct. If the message type is not programmed as 'LED' for this controller, the controller shall not take any action to update its LEDs. For LED message type, the message size always consists of 4 bytes.

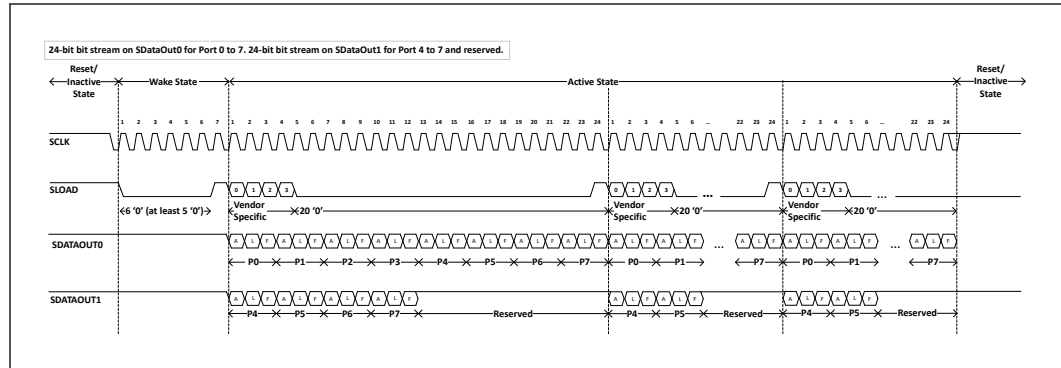
**LED Message Type**

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

the Enclosure Management LED (EM\_LED) register, Intel® Processor and Intel® Core™ i3 N-series Datasheet Volume 2 of 2 (#759604). The LEDs shall retain their values until there is a following update for that particular slot.

**SGPIO Waveform**

**Figure 20. Serial Data Transmitted over SGPIO Interface**



**20.2 Signals Description**

Name	Type	Description
GPP_E4 / <b>SATA_DEVSLP0</b>	OD	<b>Serial ATA Port [0] Device Sleep:</b> This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state. Design Constraint: no external Pull-up or Pull-down termination required when used as DEVSLP. <i>Note:</i> This pin can be mapped to SATA Port 0.
GPP_E5 / <b>SATA_DEVSLP1</b>	OD	<b>Serial ATA Port [1] Device Sleep:</b> This is an open-drain pin on the PCH side. PCH will tri-state this pin to signal to the SATA device that it may enter a lower power state (pin will go high due to Pull-up that's internal to the SATA device, per DEVSLP specification). PCH will drive pin low to signal an exit from DEVSLP state. Design Constraint: no external Pull-up or Pull-down termination required when used as DEVSLP. <i>Note:</i> This pin can be mapped to SATA Port 1. <i>Note:</i> This pin can be mapped to SATA Port 1.
PCIE11_TXN / <b>SATA0_TXN</b> PCIE11_TXP / <b>SATA0_TXP</b>	O	<b>Serial ATA Differential Transmit Pair 0:</b> These outbound SATA Port 0 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE11_RXN / <b>SATA0_RXN</b> PCIE11_RXP / <b>SATA0_RXP</b>	I	<b>Serial ATA Differential Receive Pair 0:</b> These inbound SATA Port 0 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE12_TXN / <b>SATA1_TXN</b> PCIE12_TXP / <b>SATA1_TXP</b>	O	<b>Serial ATA Differential Transmit Pair 1:</b> These outbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.
PCIE12_RXN / <b>SATA1_RXN</b> PCIE12_RXP / <b>SATA1_RXP</b>	I	<b>Serial ATA Differential Receive Pair 1:</b> These inbound SATA Port 1 high-speed differential signals support 1.5 Gb/s, 3 Gb/s and 6 Gb/s.

*continued...*

Name	Type	Description
GPP_E0 / SATAXPCE0 / <b>SATAGP0</b>	I	<b>Serial ATA Port [0] General Purpose Inputs:</b> When configured as SATAGP0, this is an input pin that is used as an interlock switch status indicator for SATA Port 0. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open. <i>Note:</i> The default use of this pin is GPP_E0. Pin defaults to Native mode as SATAXPCE0 depends on soft-strap.
GPP_A12 / SATAXPCE1 / <b>SATAGP1</b>	I	<b>Serial ATA Port [1] General Purpose Inputs:</b> When configured as SATAGP1, this is an input pin that is used as an interlock switch status indicator for SATA Port 1. Drive the pin to '0' to indicate that the switch is closed and to '1' to indicate that the switch is open. <i>Note:</i> This default use of this pin is GPP_A12. Pin defaults to Native mode as SATAXPCE1 depends on soft-strap.
GPP_B14 / SPKR / TIME_SYNC1 / <b>SATA_LED#</b> / ISH_GP6	OD	<b>Serial ATA LED:</b> This signal is an open-drain output pin driven during SATA command activity. It is to be connected to external circuitry that can provide the current to drive a platform LED. When active, the LED is on. When tri-stated, the LED is off. <i>Note:</i> An external Pull-up resistor to VCC3_3 is required.

### 20.3 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type
SATAXPCE[0:1]	Internal pull-up
<i>Note:</i> Internal Pull-Up Resistors are 20 kohm ± 30% unless specified.	

### 20.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>3</sup>	Immediately after Reset <sup>3</sup>	S3/S4/S5	Deep Sx
SATA[0:1]_TXN SATA[0:1]_TXP SATA[0:1]_RXN SATA[0:1]_RXP	Primary	Internal Pull-down	Internal Pull-down	Internal Pull-down	OFF
SATA_LED#	Primary	Undriven	Undriven	Undriven	OFF
SATA_DEVSLP[0:1] <sub>1</sub>	Primary	Undriven	Undriven	Driven Low	OFF
SATAGP[0:1] <sup>2</sup>	Primary	Undriven	Undriven	Undriven	OFF
SATAXPCE[0:1] <sup>2</sup>	Primary	Internal Pull-up	Internal Pull-up	Undriven	OFF
<p><i>Notes:</i> 1. Pin defaults to GPIO mode. The pin state during and immediately after reset follows default GPIO mode pin state. The pin state for S0 to Deep Sx reflects assumption that GPIO Use Select register was programmed to native mode functionality. If GPIO Use Select register is programmed to GPIO mode, refer to Multiplexed GPIO (Defaults to GPIO Mode) section for the respective pin states in S0 to Deep Sx.</p> <p>2. Pin defaults to Native mode as SATAXPCEIx depends on soft-strap.</p> <p>3. Reset reference for primary well pins is RSMRST#.</p>					

## 21.0 Universal Flash Storage (UFS)

The Universal Flash System (UFS) is the next generation storage standard that is to replace eMMC. UFS includes the feature set of eMMC as a subset and defines a unique feature set that provides low power consumption, high data throughput, low electromagnetic interference and optimization for mass memory subsystem efficiency.

PCH supports for UFS version 2.1 specification.

### 21.1 Functional Description

UFS adopts the latest technology from other industrial standards,

- MIPI M-PHY specification version 3.1 for physical layer
- MIPI UniPro specification version 1.61 for interconnect layer
- INCTS T10 SCSI standards (SBC, SPC and SAM) for command sets and architecture model

The UFS Host Controller handles UFS Protocol at transmission level, packing data, adding cyclic redundancy check (CRC), start/end bit, and checking for transaction format correctness.

### 21.2 Signals Description

Name	Type	Description
PCIE9_TXP / <b>UFS10_TXP</b>	O	UFS port 1 lane 0 transmit signal
PCIE9_TXN / <b>UFS10_TXN</b>	O	UFS port 1 lane 0 transmit signal
PCIE9_RXP / <b>UFS10_RXP</b>	I	UFS port 1 lane 0 receive signal
PCIE9_RXN / <b>UFS10_RXN</b>	I	UFS port 1 lane 0 receive signal
PCIE10_TXP / <b>UFS11_TXP</b>	O	UFS port 1 lane 1 transmit signal
PCIE10_TXN / <b>UFS11_TXN</b>	O	UFS port 1 lane 1 transmit signal
PCIE10_RXP / <b>UFS11_RXP</b>	I	UFS port 1 lane 1 receive signal
PCIE10_RXN / <b>UFS11_RXN</b>	I	UFS port 1 lane 1 receive signal
CLKOUT_PCIE_N4 / <b>UFS_REF_CLK</b>	O	UFS reference clock signal (19.2 MHz). <i>Note:</i> Level shifter is required for this signal to meet UFS specification. Output voltage of UFS_REF_CLK is 1.05V
<b>UFS_RESET#</b>	O	Unconnected pin (UFS device reset should be connected to a level shifted version of platform reset signal).

### 21.3 I/O Signals Planes and States

<b>Signal Name</b>	<b>Power Plane</b>	<b>During Reset</b>	<b>Immediately after Reset</b>	<b>S4/S5</b>	<b>Deep Sx</b>
UFS1x_TXP/N, UFS1x_RXP/N	Primary	Driven Low	Driven Low	Driven Low	OFF
UFS_RESET#	Primary	Driven Low	Driven High	Driven Low	OFF
UFS_REF_CLK	Primary	Undriven	Undriven	Undriven	Undriven



## 22.0 Graphics

---

### 22.1 Processor Graphics

The processor graphics is based on X<sup>e</sup> graphics core architecture that enables substantial gains in performance and lower-power consumption over prior generations. X<sup>e</sup> architecture supports up to 32 Execution Units (EUs) depending on the processor SKU.

The processor graphics architecture delivers high dynamic range of scaling to address segments spanning low power to high power, increased performance per watt, support for next generation of APIs. X<sup>e</sup> scalable architecture is partitioned by usage domains along Render/Geometry, Media, and Display. The architecture also delivers very low-power video playback and next generation analytics and filters for imaging related applications. The new Graphics Architecture includes 3D compute elements, Multi-format HW assisted decode/encode pipeline, and Mid-Level Cache (MLC) for superior high definition playback, video quality, and improved 3D performance and media.

#### 22.1.1 Media Support (Intel® QuickSync and Clear Video Technology HD)

X<sup>e</sup> implements multiple media video codecs in hardware as well as a rich set of image processing algorithms.

##### 22.1.1.1 Hardware Accelerated Video Decode

X<sup>e</sup> implements a high-performance and low-power HW acceleration for video decoding operations for multiple video codecs.

The HW decode is exposed by the graphics driver using the following APIs:

- Direct3D\* 9 Video API (DXVA2)
- Direct3D11 Video API
- Intel Media SDK
- MFT (Media Foundation Transform) filters.
- Intel VA API

X<sup>e</sup> supports full HW accelerated video decoding for AVC/HEVC/VP9/JPEG/AV1.

---

#### NOTE

DX9 is removed in the graphics driver, D3D9 will be supported with the translation layer called 9on12 which is implemented by Microsoft\* in Windows\* OS.

---

**Table 84. Hardware Accelerated Video Decoding**

Codec	Profile	Level	Maximum Resolution
MPEG2	Main	Main - 15 Mbps High - 40 Mbps	FHD
WMV9 (VC1)	Advanced Main Simple	L3 - 40 Mbps High - 20 Mbps Simple	3840x3840
AVC/H264	High		
	Main		4K
	4:2:0 8bit	L5.2 - 250 Mbps	4K @ 60
JPEG/MJPEG	Baseline	Unified level	16K x16K
HEVC/H265	Main 444 (4:2:0 4:2:2 4:4:4) 8 bit	L6.2 - Main tier and high tier	-
	Main 444 10 (4:2:0 4:2:2 4:4:4) 10 bit		4K @ 60 (4:4:4)
	Main 444 12 (4:2:0 4:2:2 4:4:4) 12 bit		4K @ 60
	SCC main SCC main 10 SCC main 444 SCC main 444 10		4K @ 60
VP9	1 (4:2:0 4:4:4 8 bit)		4K @ 60
	3 (4:2:0 4:4:4 10/12bit)	Unified level	4K @ 60
AV1	0 (4:2:0 8-bit)	L3	4K x 2K (video)
	0 (4:2:0 10-bit)		16K x 16K (still picture)

Expected performance: More than 16 simultaneous decode streams @ 1080p.

**NOTE**

Actual performance depends on the processor SKU, content bit rate, and memory frequency. Hardware decode for H264 SVC is not supported.

**22.1.1.2 Hardware Accelerated Video Encode**

Gen12 implements a low-power low-latency fixed function encoder and a high-quality customizable encoder with hardware assisted motion estimation engine which supports AVC, MPEG-2, HEVC, and VP9.

The HW encode is exposed by the graphics driver using the following APIs:

- Intel® Media SDK
- MFT (Media Foundation Transform) filters

X<sup>e</sup> supports full HW accelerated video encoding for AVC/HEVC/VP9/JPEG.

**Table 85. Hardware Accelerated Video Encode**

Codec	Profile	Level	Maximum Resolution
MPEG2	Main High	20 Mbps	1080p@30 fps
AVC/H264	High Main	L5.1	2160p(4K) @ 30
JPEG	Baseline	-	16Kx16K
HEVC/H265	Main Main10 8/10 bit Main 4:2:2 10 bit (VME) Main 4:4:4 10 bit (VDEnc) SCC Main 4:2:0 4:4:4 8/10 bit	L6 -main tier and high tier	2160p(4K)
VP9	0 (4:2:0 Chroma 8 bit) 1 (partial: 4:4:4 8 bit) 2 (partial: 4:2:0 10 bit) 3 (partial: 4:4:4 10 bit)	-	2160p(4K)

**NOTE**

Hardware encode for H264 SVC is not supported.

**22.1.1.3 Hardware Accelerated Video Processing**

There is hardware support for image processing functions such as De-interlacing, Film cadence detection, Advanced Video Scaler (AVS), detail enhancement, gamut compression, HD adaptive contrast enhancement, skin tone enhancement, total color control, Chroma de-noise, SFC (Scalar and Format Conversion), memory compression, Localized Adaptive Contrast Enhancement (LACE), spatial de-noise, 16 bpc support for de-noise/de-mosaic, Facial filter, HDR10 Tone Mapping HW acceleration.

The HW video processing is exposed by the graphics driver using the following APIs:

- Direct3D\* 9 Video API (DXVA2)
- Direct3D\* 11 Video API
- Intel One VPL
- Media Foundation Transform (MFT) filters
- Intel® Graphics Control Library (IGCL)
- Intel VA API

**NOTES**

- Not all features are supported by all the above APIs. Refer to the relevant documentation for more details.
- DX9 is removed in graphics driver, D3D9 will be supported with the translation layer called 9on12 which is implemented by Microsoft\* in Windows\* OS.

#### 22.1.1.4 Hardware Accelerated Transcoding

Transcoding is a combination of decode, video processing (optional) and encode. Using the above hardware capabilities can accomplish a high-performance transcode pipeline. There is not a dedicated API for transcoding.

The processor graphics supports the following transcoding features:

- High performance high quality flexible encoder for video editing, video archiving.
- Low-power low latency encoder for video conferencing, wireless display, and game streaming.
- Lossless memory compression for media engine to reduce media power.
- High-quality Advanced Video Scaler (AVS)
- Low power Scaler and Format Converter.

## 22.2 Platform Graphics Hardware Feature

### 22.2.1 Hybrid Graphics

Microsoft\* Windows\* 10 operating system enables the Windows\*10 Hybrid graphics framework wherein the GPUs and their drivers can be simultaneously utilized to provide users with the benefits of both performance capability of discrete GPU (dGPU) and low-power display capability of the processor GPU (iGPU). For instance, when there is a high-end 3D gaming workload in progress, the dGPU will process and render the game frames using its graphics performance, while iGPU continues to perform the display operations by compositing the frames rendered by dGPU. We recommend that OEMS should seek further guidance from Microsoft\* to confirm that the design fits all the latest criteria defined by Microsoft\* to support HG.

Microsoft\* Hybrid Graphics definition includes the following:

1. The system contains a single integrated GPU and a single discrete GPU.
2. It is a design assumption that the discrete GPU has a significantly higher performance than the integrated GPU.
3. Both GPUs shall be physically enclosed as part of the system.
  - a. Microsoft\* Hybrid DOES NOT support hot-plugging of GPUs
  - b. OEMS should seek further guidance from Microsoft\* before designing systems with the concept of hot-plugging
4. Starting with Windows\*10 Th1 (WDDM 2.0), a previous restriction that the discrete GPU is a render-only device, with no displays connected to it, has been removed. A render-only configuration with NO outputs is still allowed, just NOT required.

## 23.0 Display

### 23.1 Display Technologies Support

Technology	Standard
<b>eDP* 1.4b</b>	VESA* Embedded DisplayPort* Standard 1.4b
<b>MIPI DSI</b>	MIPI* DSI 2 Specification Version 1.0 MIPI* DPHY Specification Version 2.0
<b>DisplayPort* 1.4a</b>	VESA* DisplayPort* Standard 1.4a VESA* DisplayPort* PHY Compliance Test Specification 1.4a VESA* DisplayPort* Link Layer Compliance Test Specification 1.4 VESA* DisplayPort* Alt Mode on USB Type-C Standard Version 1.0b
<b>HDMI2.1 TMDS Compatible</b>	High-Definition Multimedia Interface Specification Version 2.0b

### 23.2 Display Interfaces

#### 23.2.1 Digital Display Interface (DDI) Signals

Signal Name	Description	Dir.	Link Type
DDIx_TXP[3:0] DDIx_TXN[3:0]	Digital Display Interface Transmitter lanes. DisplayPort, Embedded DisplayPort, HDMI and MIPI DSI Differential Pairs.	O	Diff
DDIx_AUXP DDIx_AUXN	Digital Display Interface Display Port Auxiliary: Half-duplex, bidirectional channel consist of one differential pair for each channel. MIPI DSI interface differential pair.	I/O	Diff
DISP_UTILS_1 / DSI_DE_TE_1	Digital Display Interface Utility Pin. MIPI DSI Tearing effect signal	O	SE
DISP_UTILS_2 / DSI_DE_TE_2	Digital Display Interface Utility Pin. MIPI DSI Tearing effect signal.	O	SE
DDIA_RCOMP DDIB_RCOMP	DDI IO Compensation resistors.	A	SE

*Note:* x Can be ports A, B

### 23.3 Display Configuration

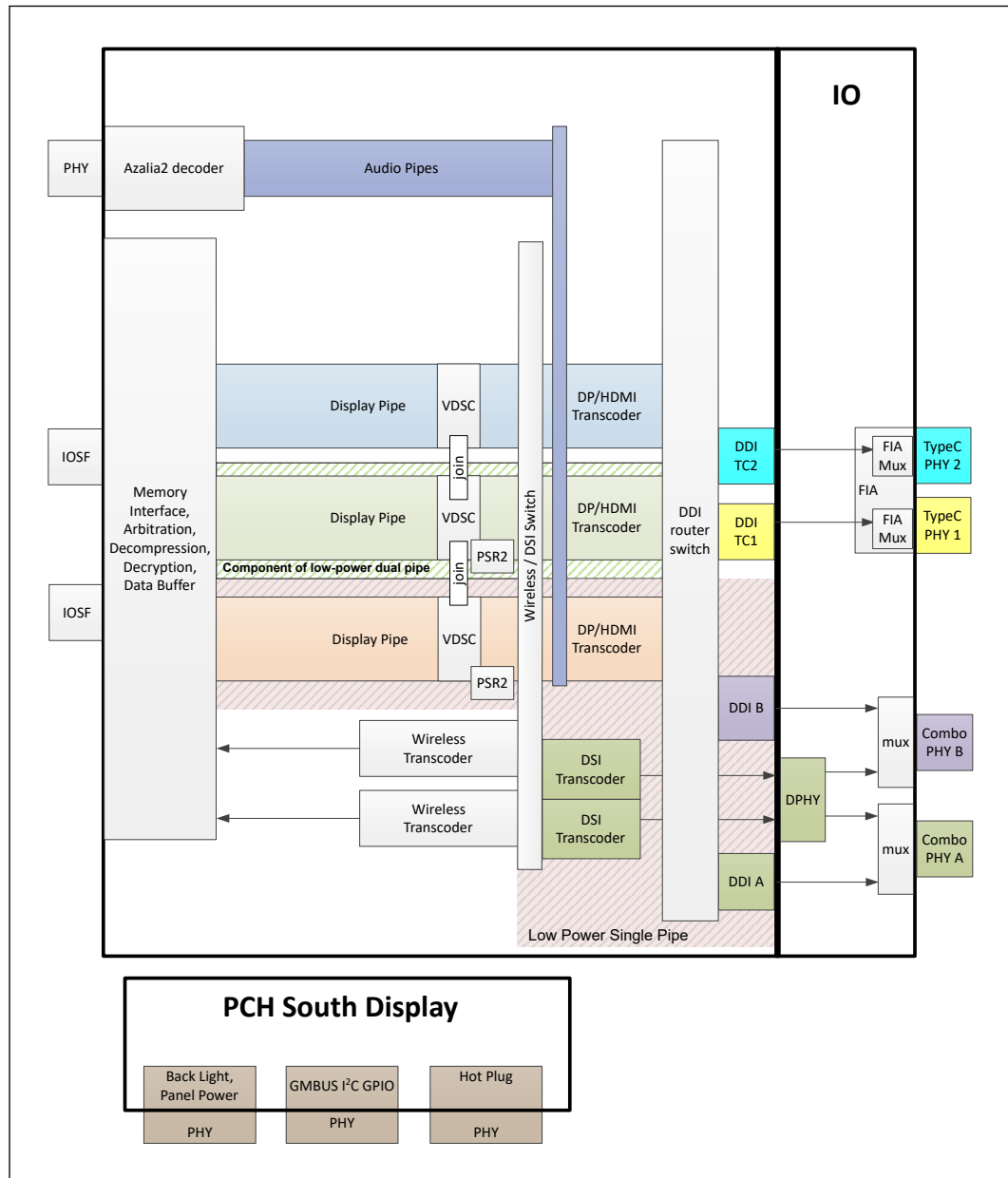
**Table 86. Display Ports Availability and Link Rate**

Port	Supported Configuration
DDI A	eDP* up to HBR3 MIPI DSI up to 2.5 Gbps

*continued...*

<b>Port</b>	<b>Supported Configuration</b>
	DP* up to HBR3 <sup>1</sup> HDMI* up to 5.94 Gbps
DDI B	DP* up to HBR3 <sup>1</sup> HDMI* up to 5.94 Gbps
TCP 0	DP* up to HBR3 HDMI* up to 5.94 Gbps
TCP 1	DP* up to HBR3 HDMI* up to 5.94 Gbps
<i>Notes:</i> 1. On board re-timer is required. 2. HBR3 - 8.1 Gbps lane rate. 3. HBR2 - 5.4 Gbps lane rate.	

Figure 21. Processor Display Architecture



## 23.4 Display Features

### 23.4.1 General Capabilities

- Up to three simultaneous displays.
  - Up to 1x4kp60 + 2x1080p60 display concurrent.
  - Up to 2x4K60 HDR
- Display interfaces supported:

- DDI interfaces supports DP\*, HDMI\*, eDP\*, DSI\*
- TCP interfaces supports DP\*, HDMI\*, Display Alt Mode over Type-C.
- Single wireless display capture.
- Audio stream support on external ports.
- HDR (High Dynamic Range) support.
- Three Display Pipes - Supporting blending, color adjustments, scaling and dithering.
- Transcoders - Containing the Timing generators supporting eDP\*, DP\*, HDMI\* interfaces.
- Up to two Low Power optimized pipes supporting Embedded DisplayPort\* and/or MIPI\* DSI.
  - LACE (Localized Adaptive Contrast Enhancement), supported up to 4 K resolutions.
  - 3D LUT - power efficient pixel modification function for color processing.
  - FBC (Frame Buffer Compression) - power saving feature.

### 23.4.2 Multiple Display Configurations

The following multiple display configuration modes are supported (with appropriate driver software):

- Single Display is a mode with one display port activated to display the output to one display device.
- Display Clone is a mode with up to three display ports activated to drive the display content of same color depth setting but potentially different refresh rate and resolution settings to all the active display devices connected.
- Extended Desktop is a mode with up to three display ports activated to drive the content with potentially different color depth, refresh rate, and resolution settings on each of the active display devices connected.

### 23.4.3 High-bandwidth Digital Content Protection (HDCP)

HDCP is the technology for protecting high-definition content against unauthorized copy or unreceptive between a source (computer, digital set top boxes, and so on) and the sink (panels, monitor, and TVs). The processor supports both HDCP 2.2 and 1.4 content protection over wired displays (HDMI\* and DisplayPort\*).

The HDCP 1.4, 2.2 keys are integrated into the processor.

### 23.4.4 DisplayPort\*

The DisplayPort\* is a digital communication interface that uses differential signaling to achieve a high-bandwidth bus interface designed to support connections between PCs and monitors, projectors, and TV displays.

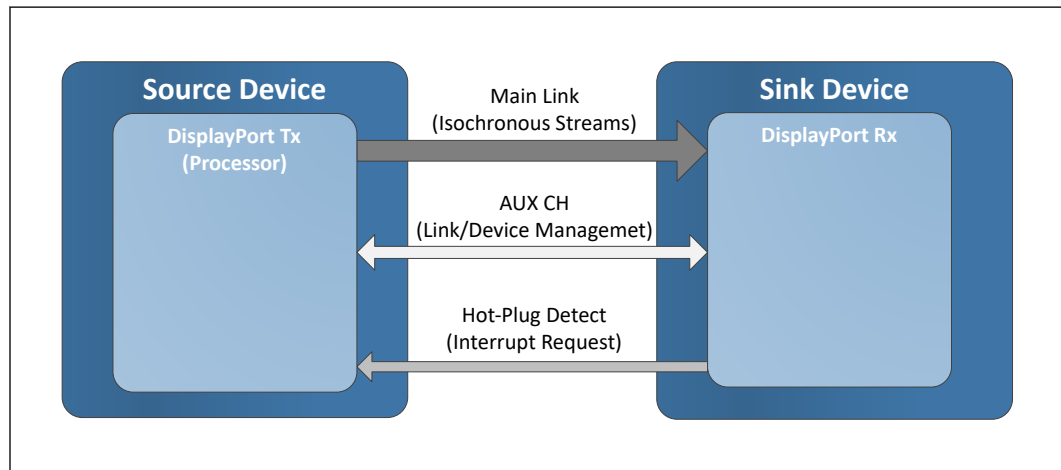
A DisplayPort\* consists of a Main Link (four lanes), Auxiliary channel, and a Hot-Plug Detect signal. The Main Link is a unidirectional, high-bandwidth, and low-latency channel used for transport of isochronous data streams such as uncompressed video



and audio. The Auxiliary Channel (AUX CH) is a half-duplex bi-directional channel used for link management and device control. The Hot-Plug Detect (HPD) signal serves as an interrupt request from the sink device to the source device.

The processor is designed in accordance with VESA\* DisplayPort\* specification. Refer to [Display Technologies Support](#) on page 197.

**Figure 22. DisplayPort\* Overview**



- Support main link of 1, 2, or 4 data lanes.
- Link rate support up to HBR3.
- Aux channel for Link/Device management.
- Hot Plug Detect.
- Support up to 36 BPP (Bit Per Pixel).
- Support SSC.
- Support YCbCR 4:4:4, YCbCR 4:2:0, YCbCR 4:2:2, and RGB color format.
- Support MST (Multi-Stream Transport).
- Support VESA DSC 1.1.
- Adaptive Sync.

**23.4.4.1 Multi-Stream Transport (MST)**

- The processor supports Multi-Stream Transport (MST), enabling multiple monitors to be used via a single DisplayPort connector.
- Maximum MST DP supported resolution:

**Table 87. Display Resolutions and Link Bandwidth for Multi-Stream Transport Calculations**

Pixels per Line	Lines	Refresh Rate [Hz]	Pixel Clock [MHz]	Link Bandwidth [Gbps]
1920	1080	60	148.5	4.46
1920	1200	60	154	4.62
<i>continued...</i>				

Pixels per Line	Lines	Refresh Rate [Hz]	Pixel Clock [MHz]	Link Bandwidth [Gbps]
2048	1152	60	156.75	4.70
2048	1280	60	174.25	5.23
2048	1536	60	209.25	6.28
2304	1440	60	218.75	6.56
2560	1440	60	241.5	7.25
3840	2160	30	262.75	7.88
2560	1600	60	268.5	8.06
2880	1800	60	337.5	10.13
3200	2400	60	497.75	14.93
3840	2160	60	533.25	16.00
4096	2160	60	556.75	16.70
4096	2304	60	605	18.15

*Notes:*

- All the above is related to bit depth of 24.
- The data rate for a given video mode can be calculated as- Data Rate = Pixel Frequency \* Bit Depth.
- The bandwidth requirements for a given video mode can be calculated as: Bandwidth = Data Rate \* 1.25 (for 8b/10b coding overhead).
- The link bandwidth depends if the standards is reduced blanking or not.  
If the standard is not reduced blanking - the expected bandwidth may be higher.  
For more details, refer to VESA and Industry Standards and Guidelines for Computer Display Monitor Timing (DMT). Version 1.0, Rev. 13 February 8, 2013
- To calculate what are the resolutions that can be supported in MST configurations, follow the below guidelines:
  - Identify what is the link bandwidth column according to the requested display resolution.
  - Summarize the bandwidth for two of three displays accordingly, and make sure the final result is below 21.6 Gbps. (for example: 4 lanes HBR2 bit rate)
 For example:
  - Docking two displays: 3840x2160@60 Hz + 1920x1200@60hz = 16 + 4.62 = 20.62 Gbps [Supported]
  - Docking three displays: 3840x2160@30 Hz + 3840x2160@30 Hz + 1920x1080@60 Hz = 7.88 + 7.88 + 4.16 = 19.92 Gbps [Supported].

**Table 88. DisplayPort Maximum Resolution**

Standard	Resolution Supported
DP*	4096x2304 60Hz 36bpp

*Notes:*

- Maximum resolution is based on the implementation of 4 lanes at HBR3 link data rate.
- bpp - bit per pixel.
- Resolution support is subject to memory BW availability.
- Resolutions will consume two display pipes.

### 23.4.5 High-Definition Multimedia Interface (HDMI\*)

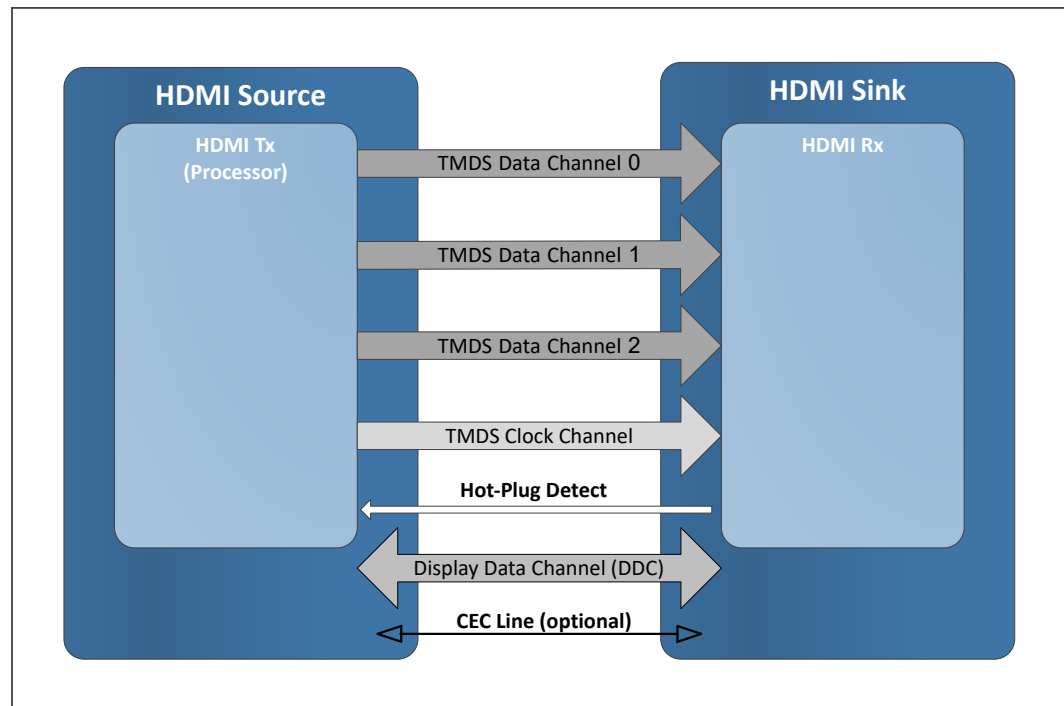
The High-Definition Multimedia Interface (HDMI\*) is provided for transmitting uncompressed digital audio and video signals from DVD players, set-top boxes, and other audio-visual sources to television sets, projectors, and other video displays. It can carry high-quality multi-channel audio data and all standard and high-definition

consumer electronics video formats. The HDMI display interface connecting the processor and display devices uses transition minimized differential signaling (TMDS) to carry audiovisual information through the same HDMI cable.

HDMI\* includes three separate communications channels: TMDS, DDC, and the optional CEC (consumer electronics control). CEC is not supported on the processor. As shown in the following figure, the HDMI\* cable carries four differential pairs that make up the TMDS data and clock channels. These channels are used to carry video, audio, and auxiliary data. In addition, HDMI carries a VESA DDC. The DDC is used by an HDMI\* Source to determine the capabilities and characteristics of the Sink.

Audio, video, and auxiliary (control/status) data is transmitted across the three TMDS data channels. The video pixel clock is transmitted on the TMDS clock channel and is used by the receiver for data recovery on the three data channels. The digital display data signals driven natively through the PCH are AC coupled and needs level shifting to convert the AC coupled signals to the HDMI\* compliant digital signals. The processor HDMI\* interface is designed in accordance with the High-Definition Multimedia Interface.

**Figure 23. HDMI\* Overview**



- DDC (Display Data Channel) channel.
- Support YCbCR 4:4:4, YCbCR 4:2:0, YCbCR 4:2:2, and RGB color format.
- Support up to 36 BPP (Bit Per Pixel).
- Hot Plug Detect.

**Table 89. HDMI Maximum Resolution**

Standard	Resolution Supported
HDMI 1.4	4Kx2K 24-30 Hz 24bpp
HDMI2.1 TMDS Compatible	4Kx2K 48-60Hz 24bpp (RGB/YUV444) 4Kx2K 48-60Hz 12bpc (YUV420)
<i>Notes:</i> 1. bpp - bit per pixel. 2. Resolution support is subject to memory BW availability.	

### 23.4.6 embedded DisplayPort\* (eDP\*)

The embedded DisplayPort\* (eDP\*) is an embedded version of the DisplayPort standard oriented towards applications such as notebook and All-In-One PCs. Like DisplayPort, embedded DisplayPort\* also consists of the Main Link, Auxiliary channel, and an optional Hot-Plug Detect signal.

- Supported on Low power optimized pipes.
- Support up to HBR3 link rate.
- Support Backlight PWM control and enable signals, and power enable.
- Support VESA DSC 1.1.
- Support SSC.
- Panel Self Refresh 1.
- Panel Self Refresh 2
- MSO 2x2 (Multi Segment Operation).
- Dedicated Aux channel.
- Adaptive Sync.

**Table 90. Embedded DisplayPort Maximum Resolution**

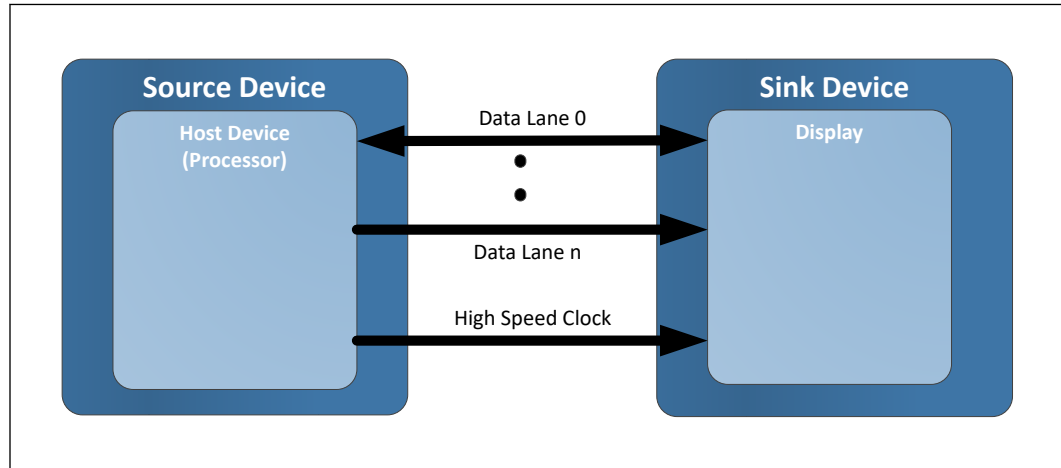
Standard	Resolution Supported
eDP*	1080p 60Hz
<i>Notes:</i> 1. Maximum resolution is based on the implementation of 4 lanes at HBR3 link data rate. 2. bpp - bit per pixel. 3. Resolution support is subject to memory BW availability. 4. High resolution are supported, validation is dependant on panel market availability.	

### 23.4.7 MIPI\* DSI

Display Serial Interface (DSI\*) specifies the interface between a host processor and peripherals such as a display module. DSI is a high speed and high performance serial interface that offers efficient and low power connectivity between the processor and the display module.

- One link x8 data lanes.
- Supported on Low power optimized pipes.
- Support Backlight PWM control and enable signals, and power enable.
- Support VESA DSC (Data Stream Compression).

**Figure 24. MIPI\* DSI Overview**



**Table 91. MIPI\* DSI Maximum Resolution**

Standard	Resolution Supported
MIPI* DSI (Single Link)	1080p 60 Hz
Notes: 1. bpp - bit per pixel. 2. Resolution support is subject to memory BW availability.	

### 23.4.8 Integrated Audio

- HDMI\* and DisplayPort interfaces can carry audio along with video.
- The processor supports three High Definition audio streams on three digital ports simultaneously (the DMA controllers are in PCH).
- The integrated audio processing (DSP) is performed by the PCH and delivered to the processor using the AUDIO\_SDI and AUDIO\_CLK inputs pins.
- The AUDIO\_SDO output pin is used to carry responses back to the PCH.
- Supports only the internal HDMI and DP CODECs.

**Table 92. Processor Supported Audio Formats over HDMI\* and DisplayPort\***

Audio Formats	HDMI*	DisplayPort*
AC-3 Dolby* Digital	Yes	Yes
Dolby* Digital Plus	Yes	Yes
DTS-HD*	Yes	Yes
LPCM, 192 kHz/24 bit, 6 Channel	Yes	Yes
Dolby* TrueHD, DTS-HD Initiator Audio* (Lossless Blu-Ray Disc* Audio Format)	Yes	Yes

The processor will continue to support Silent stream. A Silent stream is an integrated audio feature that enables short audio streams, such as system events to be heard over the HDMI\* and DisplayPort\* monitors. The processor supports silent streams over the HDMI and DisplayPort interfaces at 44.1 kHz, 48 kHz, 88.2 kHz, 96 kHz, 176.4 kHz, and 192 kHz sampling rates and silent multi-stream support.

## 24.0 High Precision Event Timer (HPET)

---

### 24.1 Feature Overview

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

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

Timer 0 supports periodic interrupts.

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

#### 24.1.1 Timer Accuracy

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

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

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

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

#### 24.1.2 Timer Off-load

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

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

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

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

### Theory of Operation

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

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

When transitioning from fast clock to slow clock, the fast clock value will be loaded into the upper 64 bits of the 82 bit counter, with the 18 LSBs set to zero. The actual transition though happens in two stages to avoid metastability. There is a fast clock sampling of the slow clock through a double flop synchronizer. Following a request to transition to the slow clock, the edge of the slow clock is detected and this causes the fast clock value to park. At this point the fast clock can be gated. On the next rising edge of the slow clock, the parked fast clock value (in the upper 64 bits of an 82 bit value) is added to the value from the Off-load Calibrator. On subsequent edges while in slow clock mode the slow clock counter increments its count by the value from the Off-load Calibrator.

When transitioning from slow clock to fast clock, the fast clock waits until it samples a rising edge of the slow clock through its synchronizer and then loads the upper 64 bits of the slow clock value as the fast count value. It then de-asserts the indication that slow clock mode is active. The 32 kHz clock counter no longer counts. The 64 bit MSB will be over-written when the 32 kHz counter is reloaded once conditions are met to enable the 32 kHz HPET counter but the 18 bit LSB is retained and it is not cleared out during the next reload cycle to avoid losing the fractional part of the counter.

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

### 24.1.3 Interrupt Mapping

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

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

In this case, the Legacy Replacement Rout bit (LEG\_RT\_CNF) is set. This forces the mapping found in below table.

**Table 93. Legacy Replacement Routing**

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

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

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

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

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

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

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

---

#### NOTE

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

---



---

**NOTE**

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

---

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

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

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

---

**NOTE**

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

---

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

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

## 24.1.4 Periodic Versus Non-Periodic Modes

### Non-Periodic Mode

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

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

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

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

- Set TIMER0\_VAL\_SET\_CNF bit
- Set the lower 32 bits of the Timer0 Comparator Value register
- Set TIMER0\_VAL\_SET\_CNF bit
- Set the upper 32 bits of the Timer0 Comparator Value register

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

---

**WARNING**

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

---

All of the timers support non-periodic mode.

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

**Periodic Mode**

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

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

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

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

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

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

Software sets the `ENABLE_CNF` bit to enable interrupts.

---

**NOTE**

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

---

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

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

### 24.1.5 Enabling the Timers

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

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

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

### 24.1.6 Interrupt Levels

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

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

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

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

## 25.0 8254 Timers

---

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

### Counter 0, System Timer

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

### Counter 2, Speaker Tone

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

## 25.1 Timer Programming

The counter/timers are programmed in the following fashion:

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

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

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

If a counter is programmed to read/write 2-byte counts, the following precaution applies – a program must not transfer control between writing the first and second Byte to another routine, which also writes into that same counter. Otherwise, the counter will be loaded with an incorrect count.

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

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

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

**Table 94. 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.

## 25.2 Reading from the Interval Timer

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

With the simple read and counter latch command methods, the count must be read according to the programmed format; specifically, if the counter is programmed for 2-byte counts, 2-bytes must be read. The 2-bytes do not have to be read one right after the other. Read, write, or programming operations for other counters may be inserted between them.

### Simple Read

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

---

**NOTE**

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

---

**Counter Latch Command**

The Counter Latch command, written to Port 43h, latches the count of a specific counter at the time the command is received. This command is used to ensure that the count read from the counter is accurate, particularly when reading a 2-byte count. The count value is then read from each counter's Count register as was programmed by the Control register.

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

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

**Read Back Command**

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

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

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

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

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

## 26.0 Processor Sideband Signals

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

Acronyms	Description
PECI	Platform Environmental Control Interface

### 26.1 Functional Description

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

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

### 26.2 Signal Description

Name	Type	Description
<b>PROCPWRGD</b>	O	Signal to the processor to indicate its primary power is good.
<b>THRMTRIP#</b>	I	Signal from the processor to indicate that a thermal overheating has occurred.
<b>PECI</b>	I/O	Single-wire serial bus for accessing processor digital thermometer
GPP_E3 / <b>PROC_GP0</b>	I	Thermal management signal
GPP_E7 / <b>PROC_GP1</b>	I	Thermal management signal
GPP_B3 / <b>PROC_GP2</b> / ISH_GP4B	I	Thermal management signal
GPP_B4 / <b>PROC_GP3</b> / ISH_GP5B	I	Thermal management signal

### 26.3 Integrated Pull-Ups and Pull-Downs

None

### 26.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>PROCPWRGD</b>	Primary	Undriven	Driven High	OFF	OFF
<b>THRMTRIP#</b>	Primary	Undriven	Undriven	OFF	OFF

*continued...*

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>PECI</b>	Primary	Undriven	Undriven	OFF	OFF
<b>PROC_GP[3:0]</b>	Primary	Undriven	Undriven	Undriven	OFF
<i>Note:</i> 1. Reset reference for primary well pins is RSMRST#.					



## 27.0 General Purpose Input and Output

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

The high level features of GPIO:

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

**Table 95. 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

### 27.1 Functional Description

This section provides information on the following topics:

- Configurable GPIO Voltage
- GPIO Buffer Impedance Compensation
- Interrupt / IRQ via GPIO Requirement
- Programmable Hardware Debouncer
- Integrated Pull-ups and Pull-downs
- SCI / SMI# and NMI
- Timed GPIO (TIME\_SYNC)
- GPIO Blink (BK) and Serial Blink (SBK)
- GPIO Ownership
- Native Function and TERM Bit Setting

### 27.1.1 Configurable GPIO Voltage

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

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

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

---

#### WARNING

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

---

---

#### NOTES

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

### 27.1.2 GPIO Buffer Impedance Compensation

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

### 27.1.3 Interrupt / IRQ via GPIO Requirement

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

### 27.1.4 Programmable Hardware Debouncer

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

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

### 27.1.5 Integrated Pull-ups and Pull-downs

All GPIOs have programmable internal pull-up/pull-down resistors which are off by default. The internal pull-up/pull-down for each GPIO can be enabled by BIOS programming the corresponding PAD\_CFG\_DW1 register. Refer to Volume 2 (Register Information) for more details.

### 27.1.6 SCI / SMI# and NMI

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

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

- GPP\_B14 and GPP\_B23
- GPP\_C[23:22]
- GPP\_D[4:0]
- GPP\_E[8:0] ; GPP\_E[16:13]
- GPP\_F12

### 27.1.7 Timed GPIO

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

Timed GPIO can be an input or an output:

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

---

#### NOTE

TIME\_SYNC can be set as input when both Direction (DIR) bit and Enable (EN) bit in Timed GPIO Control Register are set to 1 (refer to Intel® Processor and Intel® Core™ i3 N-series Datasheet Volume 2 of 2 (#759604) for the register information). When EN bit is set to 0, TIME\_SYNC will default to output low regardless of DIR bit setting.

---

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

---

#### NOTE

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

---

### 27.1.8 GPIO Blink (BK) and Serial Blink (SBK)

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

### 27.1.9 GPIO Ownership

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

### 27.1.10 Native Function and TERM Bit Setting

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

**Table 96. Native Function Signals Supporting Dynamic Termination Override**

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

## 27.2 Signal Description

For GPIO pin implementation including multiplexed native functions, default values, signal states, and other characteristics, refer to the download the pdf, click on the navigation pane and refer the spreadsheet, **759603-001\_GPIO.xlsx**.

## 28.0 GPIO Serial Expander

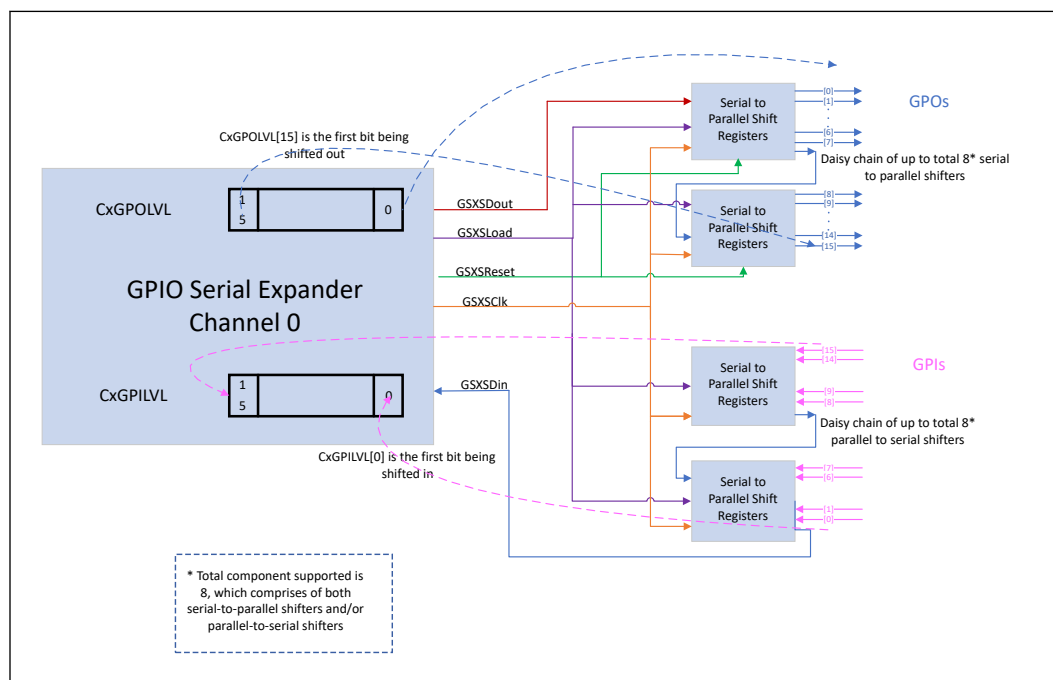
GPIO Serial Expander (GSX) is the capability provided by the PCH to expand the GPIOs on a platform that needs more GPIOs than the ones provided by the PCH. The solution requires external shift register discrete components.

### 28.1 Functional Description

GPIO Serial Expander (GSX) uses serial-to-parallel or parallel-to-serial shift register discrete components to increase number of the GPIO pins for system use. It expands in the multiples of 8 for input or output with 8 pins per expander. The total shift register component supported is 8, which can expand the GPIOs by up to 64.

The below figure illustrates a GPIO expansion topology with 16 GPIs and 16 GPOs.

Figure 25. 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.

## 28.2 Signal Description

Name	Type	Description
GPP_F12 / <b>GSXDOUT</b> / THC1_SPI2_IO0 / GSPi1_MOSI / I2C1A_SCL	O	GPIO Serial Expander Controller Data Out
GPP_F13 / <b>GSXSLOAD</b> / THC1_SPI2_IO1 / GSPi1_MISO / I2C1A_SDA	O	GPIO Serial Expander Controller Serial Load
GPP_F14 / <b>GSXDIN</b> / THC1_SPI2_IO2	I	GPIO Serial Expander Controller Data In
GPP_F15 / <b>GSXSRESET#</b> / THC1_SPI2_IO3	O	GPIO Serial Expander Controller Serial Reset
GPP_F16 / <b>GSXCLK</b> / THC1_SPI2_CS# / GSPi1_CS0#	O	GPIO Serial Expander Controller Clock

## 28.3 Integrated Pull-ups and Pull-downs

None

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

---

The PCH implements seven I<sup>2</sup>C controllers for seven independent I<sup>2</sup>C interfaces, I2C0-I2C5 and I2C6B. Each interface is a two-wire serial interface consisting of a serial data line (SDA) and a serial clock (SCL).

I2C6B only implements the I<sup>2</sup>C host controllers and does not incorporate a DMA controller. Therefore, I2C6B is restricted to operate in PIO mode only.

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

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

---

### NOTES

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

**Table 97. Acronyms**

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

**Table 98. References**

Specification	Location
The I <sup>2</sup> C Bus Specification, Version 5	<a href="http://www.nxp.com/documents/user_manual/UM10204.pdf">www.nxp.com/documents/user_manual/UM10204.pdf</a>

## 29.1 Functional Description

This section provides information on the following topics:

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

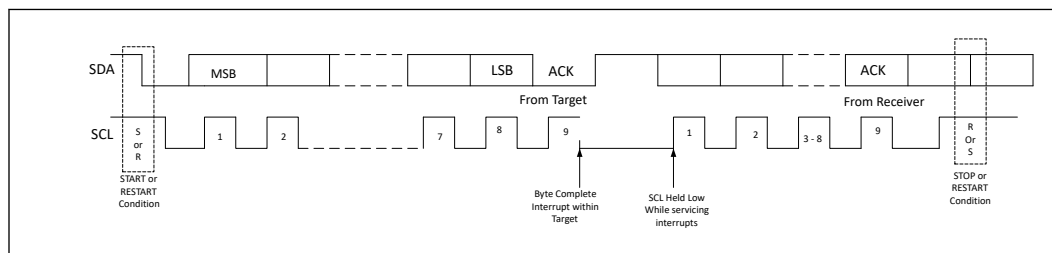
### 29.1.1 Protocols Overview

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

- The initiator generates a START condition, signaling all devices on the bus to listen for data.
- The initiator 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 initiator 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 initiator. These messages are specific to the I<sup>2</sup>C device used. After 8 message bits are written to the bus, the transmitter will receive an acknowledge bit. This message and acknowledge transfer continues until the entire message is transmitted.
- The message is terminated by the initiator with a STOP condition. This frees the bus for the next initiator to begin communications. When the bus is free, both data and clock lines are high.



**Figure 26. Data Transfer on the I<sup>2</sup>C Bus**



### Combined Formats

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

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

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

## 29.1.2 DMA Controller

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

### DMA Transfer and Setup Modes

The DMA can operate in the following modes:

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

The DMA supports the following modes for programming:

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

### Channel Control

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

### 29.1.3 Reset

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

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

---

#### NOTE

To avoid a potential I<sup>2</sup>C peripheral deadlock condition where the reset goes active in the middle of a transaction, the I<sup>2</sup>C controller must be idle before a reset can be initiated.

---

### 29.1.4 Power Management

#### Device Power Down Support

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

#### Latency Tolerance Reporting (LTR)

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

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

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

### 29.1.5 Interrupts

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

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

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

### 29.1.6 Error Handling

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

### 29.1.7 I<sup>2</sup>C Setup/Hold Time

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

## 29.2 Signal Description

Name	Type	Description
GPP_H4 / I2C0_SDA	I/OD	<b>I<sup>2</sup>C Link 0 Serial Data Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_H5 / I2C0_SCL	I/OD	<b>I<sup>2</sup>C Link 0 Serial Clock Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_E12 / THC0_SPI1_IO1 / I2C0A_SDA / GSPI0_MISO	I/OD	<b>I<sup>2</sup>C Link 0A Serial Data Line</b> External Pull-up resistor may be required depending on Bus Capacitance. Note : Alternate interface from/to the same I2C0 controller, to support touch device interface convergence.
GPP_E13 / THC0_SPI1_IO0 / I2C0A_SCL / GSPI0_MOSI	I/OD	<b>I<sup>2</sup>C Link 0A Serial Clock Line</b> External Pull-up resistor may be required depending on Bus Capacitance. Note : Alternate interface from/to the same I2C0 controller, to support touch device interface convergence.
GPP_H6 / I2C1_SDA	I/OD	<b>I<sup>2</sup>C Link 1 Serial Data Line</b>

*continued...*

Name	Type	Description
		External Pull-up resistor may be required depending on Bus Capacitance.
GPP_H7 / <b>I2C1_SCL</b>	I/OD	<b>I<sup>2</sup>C Link 1 Serial Clock Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_F13 / GSXSLOAD / THC1_SPI2_IO1 / GSPI1_MISIO / <b>I2C1A_SDA</b>	I/OD	<b>I<sup>2</sup>C Link 1A Serial Data Line</b> External Pull-up resistor may be required depending on Bus Capacitance. Note : Alternate interface from/to the same I2C1 controller, to support touch device interface convergence.
GPP_F12 / GSXDOUT / THC1_SPI2_IO0 / GSPI1_MOSI / <b>I2C1A_SCL</b>	I/OD	<b>I<sup>2</sup>C Link 1A Serial Clock Line</b> External Pull-up resistor may be required depending on Bus Capacitance. Note : Alternate interface from/to the same I2C1 controller, to support touch device interface convergence.
GPP_B5 / ISH_I2C0_SDA / <b>I2C2_SDA</b>	I/OD	<b>I<sup>2</sup>C Link 2 Serial Data Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_B6 / ISH_I2C0_SCL / <b>I2C2_SCL</b>	I/OD	<b>I<sup>2</sup>C Link 2 Serial Clock Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_B7 / ISH_I2C1_SDA / <b>I2C3_SDA</b>	I/OD	<b>I<sup>2</sup>C Link 3 Serial Data Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_B8 / ISH_I2C1_SCL / <b>I2C3_SCL</b>	I/OD	<b>I<sup>2</sup>C Link 3 Serial Clock Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_H8 / <b>I2C4_SDA</b> / CNV_MFUART2_RXD	I/OD	<b>I<sup>2</sup>C Link 4 Serial Data Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_H9 / <b>I2C4_SCL</b> / CNV_MFUART2_TXD	I/OD	<b>I<sup>2</sup>C Link 4 Serial Clock Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_D13 / ISH_UART0_RXD / <b>I2C6B_SDA</b>	I/OD	<b>2<sup>nd</sup> instance of the I<sup>2</sup>C Link 4 Serial Data Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_D14 / ISH_UART0_TXD / <b>I2C6B_SCL</b>	I/OD	<b>2<sup>nd</sup> instance of the I<sup>2</sup>C Link 4 Serial Clock Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_B16 / <b>I2C5_SDA</b> / ISH_I2C2_SDA	I/OD	<b>I<sup>2</sup>C Link 5 Serial Data Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_B17 / <b>I2C5_SCL</b> / ISH_I2C2_SCL	I/OD	<b>I<sup>2</sup>C Link 5 Serial Clock Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_H12 / <b>I2C7_SDA</b> / UART0_RTS# / SATA_DEVSLP0#	I/OD	<b>I<sup>2</sup>C Link 7 Serial Data Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_H13 / <b>I2C7_SCL</b> / UART0_CTS# / SATA_DEVSLP1#	I/OD	<b>I<sup>2</sup>C Link 7 Serial Clock Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_D16 / ISH_UART0_CTS# / SML0BALERTB / <b>I2C7B_SCL</b>	I/OD	<b>2<sup>nd</sup> instance of the I2C Link 7 Serial Clock Line</b> External Pull-up resistor may be required depending on Bus Capacitance.
GPP_D15 / ISH_UART0_RTS# / <b>I2C7B_SDA</b>	I/OD	<b>2<sup>nd</sup> instance of the I2C Link 7 Serial Data Line</b> External Pull-up resistor may be required depending on Bus Capacitance.

## 29.3 Integrated Pull-Ups and Pull-Downs

None.

## 29.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
I2C[5:0]_SDA, I2C7_SDA, I2C[1:0]_SDA, I2C6B_SDA	Primary	Undriven	Undriven	Undriven	OFF
I2C[5:0]_SCL, I2C7_SCL, I2C[1:0]_SCL, I2C6B_SCL	Primary	Undriven	Undriven	Undriven	OFF

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

## 30.0 Connectivity Integrated (CNVi)

Connectivity Integrated (CNVi) is a general term referring to a family of connectivity solutions which are based on the Connectivity Controller family. The common component of all these solutions is the Connectivity Controller IP, which is a hard macro embedded in various Intel processor chips.

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

- The containing chip (Processor or PCH which contains the Connectivity Controller IP)
- Buttress (as applicable to each platform, and coupled the Connectivity Controller IP)
- Companion RF chip that is in a pre-certified module (i.e., M.2-2230, M.2-1216) or soldered as chip on board.

The main blocks of the integrated Connectivity solution are partitioned according to the following:

**Table 99. Acronyms**

Acronyms	Description
BRI	Bluetooth* Radio Interface
CNVi	Connectivity Integrated
PCH	Platform Controller Hub
RGI	Radio Generic interface
SoC	System On Chip
IP	Literally, Intellectual Property. IP refers to architecture, design, validation, and software components collectively delivered to enable one or more specific Processor features
MFUART	Multifunction Universal Asynchronous Receiver/Transmitter
UART	Universal Asynchronous Receiver/Transmitter

**Table 100. References**

Specification	Location
M.2 Specification	<a href="https://pcisig.com/specifications/pciexpress/M.2_Specification/">https://pcisig.com/specifications/pciexpress/M.2_Specification/</a>
MIPI* Alliance specification for D-PHY v1.2	<a href="http://www.mipi.org/specifications">http://www.mipi.org/specifications</a>
Intel® Wi-Fi 6E AX210 (Typhoon Peak 2) 2D/3D 1216 Drawings and IGS file	626868

### 30.1 Functional Description

The main blocks of the integrated Connectivity solution are partitioned according to the following:

- **Connectivity Controller IP** contains:
  - Interfaces to the PCH
  - Debug and testing interfaces
  - Power management and clock Interfaces
  - Interface to the Companion RF module (CRF)
  - Interface to physical I/O pins controlled by the PCH
  - Interfaces to the LTE modem via PCH GPIO
- **Companion RF (CRF):** This is the integrated connectivity M.2 module. The CRF Top contains:
  - Debug and testing interfaces
  - Power and clock Interfaces
  - Interface to the Connectivity Controller chip
- **Physical I/O Pins:** The SCU units are responsible for generating and controlling the power and clock resources of Connectivity Controller and CRF. There are unique SCUs in Connectivity Controller and CRF and their operation is coordinated due to power and clock dependencies. This coordination is achieved by signaling over a control bus (AUX) connecting Connectivity Controller and CRF.

Both Connectivity Controller and CRF have a dedicated AUX bus and arbiter. These two AUX buses are connected by a special interface that connects over the RGI bus. Each of the Connectivity Controller and CRF cores is dedicated to handle a specific connectivity function (Wi-Fi, Bluetooth).

Only the digital part of the connectivity function is located in Connectivity Controller cores, while the CRF cores handle some digital, but mostly analog and RF functionality. Each core in the Connectivity Controller has an interface to the host and an interface to its counterpart in CRF. CRF cores include an analog part which is connected to board level RF circuitry and to an antenna.

## 30.2 Signal Description

Name	Type	Description
<b>GPIO Fixed Functions (Signals for Integrated Connectivity (CNVi) and Discrete Connectivity (CNVd) functions)</b>		
GPP_R4 / HDA_RST# / <b>I2S2_SCLK</b> / DMIC_CLK_A_0A	I/O	For CNVi: Unused For discrete connectivity with UART host support: Optional Bluetooth* I2S bus clock
GPP_F4 / <b>CNV_RF_RESET#</b>	I/O	For CNVi: RF companion (CRF) reset signal, active low. Require a 75 kohm Pull-Down on platform/motherboard level. It is recommended not to use it for bootstrapping during early Platform init flows.
GPP_R6 / <b>I2S2_TXD</b> / DMIC_CLK_1A	O	For CNVi: Unused For discrete connectivity with UART host Bluetooth* support: Optional Bluetooth* I2S bus data output (input to Bluetooth* module)
GPP_R7 / <b>I2S2_RXD</b> / DMIC_DATA_1A	I	For CNVi: Unused. For discrete connectivity with UART host support: Optional Bluetooth* I2S bus data output (from Bluetooth* module)
GPP_F0 / <b>CNV_BRI_DT</b> / UART2_RTS#	O	For CNVi: BRI bus TX.
<i>continued...</i>		

Name	Type	Description
		For discrete connectivity with UART host support: Bluetooth* UART RTS#
GPP_F1 / <b>CNV_BRI_RSP</b> / UART2_RXD	I	For CNVi: BRI bus RX. For discrete connectivity with UART host support: Bluetooth* UART RXD
GPP_F2 / <b>CNV_RGI_DT</b> / UART2_TXD	O	For CNVi: RGI bus TX. RGI_DT is used by the platform to strap presence of the CRF. Requires weak pull up of 20Kohm on the platform. For discrete connectivity with UART host support: Bluetooth* UART TXD
GPP_F3 / <b>CNV_RGI_RSP</b> / UART2_CTS#	I	For CNVi: RGI bus RX. For discrete connectivity with UART host support: Bluetooth* UART CTS#
GPP_F5 / MODEM_CLKREQ / <b>CRF_XTAL_CLKREQ</b>	O	For CNVi: Processor to CRF wake indication
GPP_F6 / <b>CNV_PA_BLANKING</b>	I/O	For CNVi and discrete connectivity : Optional WLAN/Bluetooth* WWAN co-existence signal. Used to be co-existence signal for external GNSS solution
GPP_H8 / I2C4_SDA / <b>CNV_MFUART2_RXD</b>	I	For CNVi and discrete connectivity: Optional WLAN/Bluetooth* WWAN co-existence signal (Input)
GPP_H9 / I2C4_SCL / <b>CNV_MFUART2_TXD</b>	O	For CNVi and discrete connectivity : Optional WLAN/Bluetooth* WWAN co-existence signal (Output)
<b>Fixed Special Purpose I/O</b>		
<b>CNV_WT_CLKP</b>	O	CNVio bus TX CLK+
<b>CNV_WT_CLKN</b>	O	CNVio bus TX CLK-
<b>CNV_WT_D0P</b>	O	CNVio bus Lane 0 TX+
<b>CNV_WT_D0N</b>	O	CNVio bus Lane 0 TX-
<b>CNV_WT_D1P</b>	O	CNVio bus Lane 1 TX+
<b>CNV_WT_D1N</b>	O	CNVio bus Lane 1 TX-
<b>CNV_WR_CLKP</b>	I	CNVio bus RX CLK+
<b>CNV_WR_CLKN</b>	I	CNVio bus RX CLK-
<b>CNV_WR_D0P</b>	I	CNVio bus Lane 0 RX+
<b>CNV_WR_D0N</b>	I	CNVio bus Lane 0 RX-
<b>CNV_WR_D1P</b>	I	CNVio bus Lane 1 RX+
<b>CNV_WR_D1N</b>	I	CNVio bus Lane 1 RX-
<b>Selectable Special Purpose I/O</b>		
<b>PCIE12_TXP</b>	O	Wi-Fi* PCIe* host bus TX (positive) for discrete connectivity. Optional to connect to a Wi-Fi* PCIe* PERp0 pin on the Wi-Fi* module. This is the recommended port but other PCIe* ports can be selected for this function.
<b>PCIE12_TXN</b>	O	Wi-Fi* PCIe* host bus TX (negative) for discrete connectivity. Optional to connect to a Wi-Fi* PCIe* PERn0 pin on the Wi-Fi* module. This is the recommended port but other PCIe* ports can be selected for this function.
<b>continued...</b>		



Name	Type	Description
PCIE12_RXP	I	Wi-Fi* PCIe* host bus RX (positive) for discrete connectivity. Optional to connect to a Wi-Fi* PCIe* PETp0 pin on the Wi-Fi* module. This is the recommended port but other PCIe* ports can be selected for this function.
PCIE12_RXN	I	Wi-Fi* PCIe* host bus RX (negative) for discrete connectivity. Optional to connect to a Wi-Fi* PCIe* PETn0 pin on the Wi-Fi* module. This is the recommended port but other PCIe* ports can be selected for this function.
CLKOUT_PCIE_P3	O	Wi-Fi* PCIe* host bus clock (positive) for discrete connectivity. Optional to connect to a Wi-Fi* PCIe* REFCLKp pin on the Wi-Fi* module. This is the recommended clock signal but other PCIe* clocks can be selected for this function.
CLKOUT_PCIE_N3	O	Wi-Fi* PCIe* host bus clock (negative) for discrete connectivity. Optional to connect to a Wi-Fi* PCIe* REFCLKp pin on the Wi-Fi* module. This is the recommended clock signal but other PCIe* clocks can be selected for this function.
W_Disable1# (GPIO)	O	Used for Wi-Fi* RF-Kill control. This pin can be connected to a platform switch or to Processor GPIOs (recommendation- if possible do not use GPIOs that have Platform impact as "bootstraps" during platform init). The signal must keep value in Sx state (configured in BIOS) The W_Disable signal have a Pull-up embedded in the CRF silicon, (this is an Active-Low signal).
W_Disable2# (GPIO)	O	Used for Bluetooth* RF-Kill control. This pin can be connected to a platform switch or to Processor GPIOs (recommendation- if possible do not use GPIOs that have Platform impact as "bootstraps" during platform init). The signal must keep value in Sx state (configured in BIOS) The W_Disable signal have a Pull-up embedded in the CRF silicon, (this is an Active-Low signal).

### 30.3 Integrated Pull-ups and Pull-downs

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

### 30.4 I/O Signal Planes and States

Signal Name	Power plane	During Reset <sup>1</sup>	Immediately After Reset <sup>1</sup>	S3/S4/S5	Deep Sx
CNV_RF_RESET#	Primary	Driven	Driven	Driven	OFF
CNV_MFUART2_RXD	Primary	Undriven	Undriven	Undriven	OFF
CNV_MFUART2_TXD	Primary	Undriven	Undriven	Undriven	OFF
CNV_BRI_DT	Primary	Driven	Driven	Driven	OFF
CNV_BRI_RSP	Primary	Powered (input, PU)	Powered (input, PU)	Powered (input, PU)	OFF
CNV_RGI_DT	Primary	Driven	Driven	Driven	OFF

*continued...*

Signal Name	Power plane	During Reset <sup>1</sup>	Immediately After Reset <sup>1</sup>	S3/S4/S5	Deep Sx
CNV_RGI_RSP	Primary	Powered (input, PU)	Powered (input, PU)	Powered (input, PU)	OFF
CNV_WT_CLKP	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_CLKN	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_D0P	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_D0N	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_D1P	Primary	Undriven	Undriven	Driven	OFF
CNV_WT_D1N	Primary	Undriven	Undriven	Driven	OFF
CNV_WR_CLKP	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_CLKN	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_D0P	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_D0N	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_D1P	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WR_D1N	Primary	Undriven	Undriven	Powered (input)	OFF
CNV_WT_RCOMP	Primary	Undriven	Undriven	Driven	OFF

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

## 31.0 Integrated Sensor Hub (ISH)

**Table 101. Acronyms**

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

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

### 31.1 Feature Overview

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

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

The ISH consists of the following key components:

- A combined cache for instructions and data.
  - ROM space intended for the bootloader.
  - SRAM space for code and data.
- Interfaces to sensor peripherals (I<sup>2</sup>C, UART, SPI, GPIO).
- An interface to main memory.
- Out of Band signals for clock and wake-up control.

- Inter Process Communications to the Host and Intel® CSE.
- Part of the PCI tree on the host.

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

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

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

- Initiator Mode Only (all peripherals must be target devices)
- Support for the following operating speeds:
  - Standard mode: 100 kbps
  - Fast Mode: 400 kbps
  - Fast Mode Plus: 1 000 kbps
  - High Speed Mode: 2400 kbps
- Support for both 7-bit and 10-bit addressing formats on the I<sup>2</sup>C bus
- FIFO of 64 bytes with programmable watermarks/thresholds

### 31.1.2 ISH UART Controller

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

The UART has the following capabilities:

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

### 31.1.3 ISH GSPI Controller

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

The SPI controller includes:

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

### 31.1.4 ISH GPIOs

The ISH supports eight dedicated GPIOs.

## 31.2 Functional Description

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

### 31.2.1 ISH Micro-Controller

The ISH is operated by a micro-controller. This core provides localized sensor aggregation and data processing, thus off loading the processor and lowering overall platform average power. The core supports an in-built local APIC that receives messages from the IOAPIC. A local boot ROM with FW for initialization is also part of the core.

### 31.2.2 SRAM

The local SRAM is used for ISH FW code storage and to read/write operational data. The local SRAM block includes both the physical SRAM as well as the controller logic. The SRAM is a total of 640K bytes organized into banks of 32 kB each and is 32-bit wide. The SRAM is shared with Intel® CSE 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.

### 31.2.3 PCI Host Interface

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

#### MMIO Space

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

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

#### DMA Controller

The DMA controller supports up to 64-bit addressing.

#### PCI Interrupts

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

#### PCI Power Management

PME is not supported in ISH.

### 31.2.4 Power Domains and Management

#### ISH Power Management

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

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

#### External Sensor Power Management

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

### 31.2.5 ISH IPC

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

**Table 102. IPC Initiator -> Target Flows**

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

### 31.2.6 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.

### 31.3 Signal Description

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

### 31.4 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
<b>ISH_SPI_MISO</b>	Pull-Down	20 kohm ± 30%	

### 31.5 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
ISH_I2C0_SDA	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C0_SCL	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C1_SDA	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C1_SCL	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C2_SDA	Primary	Undriven	Undriven	Undriven	OFF
ISH_I2C2_SCL	Primary	Undriven	Undriven	Undriven	OFF
ISH_GP[7:0]	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART0_TXD	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART0_RXD	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART0_RTS#	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART0_CTS#	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART1_TXD	Primary	Undriven	Undriven	Undriven	OFF
ISH_UART1_RXD	Primary	Undriven	Undriven	Undriven	OFF
ISH_SPI_CS#	Primary	Undriven	Undriven	Undriven	OFF
ISH_SPI_CLK	Primary	Undriven	Undriven	Undriven	OFF
ISH_SPI_MISO	Primary	Undriven	Undriven	Undriven	OFF
ISH_SPI_MOSI	Primary	Undriven	Undriven	Undriven	OFF
<i>Note:</i> 1. Reset reference for primary well pins is RSMRST#.					



## 32.0 System Management

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

The following features and functions are supported by the PCH:

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

**Table 103. Acronyms**

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

### 32.1 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.

#### 32.1.1 Handling an Intruder

The PCH has an input signal, INTRUDER#, that can be attached to a switch that is activated when the system's case is open. This input has a two RTC clock debounce. If INTRUDER# goes active (after the debouncer), this will set the INTRD\_DET bit in the TCO2\_STS register. The INTRD\_SEL bits in the TCO\_CNT register can enable the PCH to cause an SMI# or interrupt. The BIOS or interrupt handler can then cause a transition to the S5 state by writing to the SLP\_EN bit.

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

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

**NOTE**

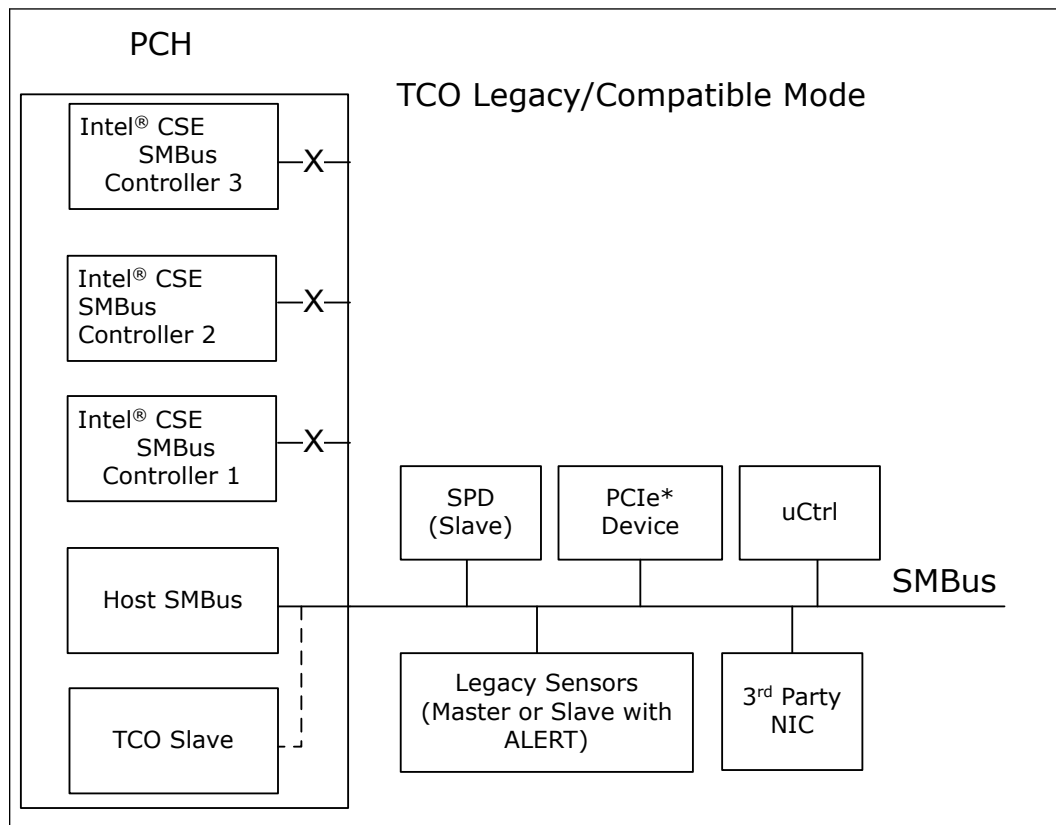
This is slightly different than a classic sticky bit, since most sticky bits would remain active indefinitely when the signal goes active and would immediately go inactive when a 1 is written to the bit.

### 32.1.2 TCO Modes

#### TCO Compatible Mode

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

**Figure 27. TCO Compatible Mode SMBus Configuration**



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

**Table 104. Event Transitions that Cause Messages**

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

### Advanced TCO Mode

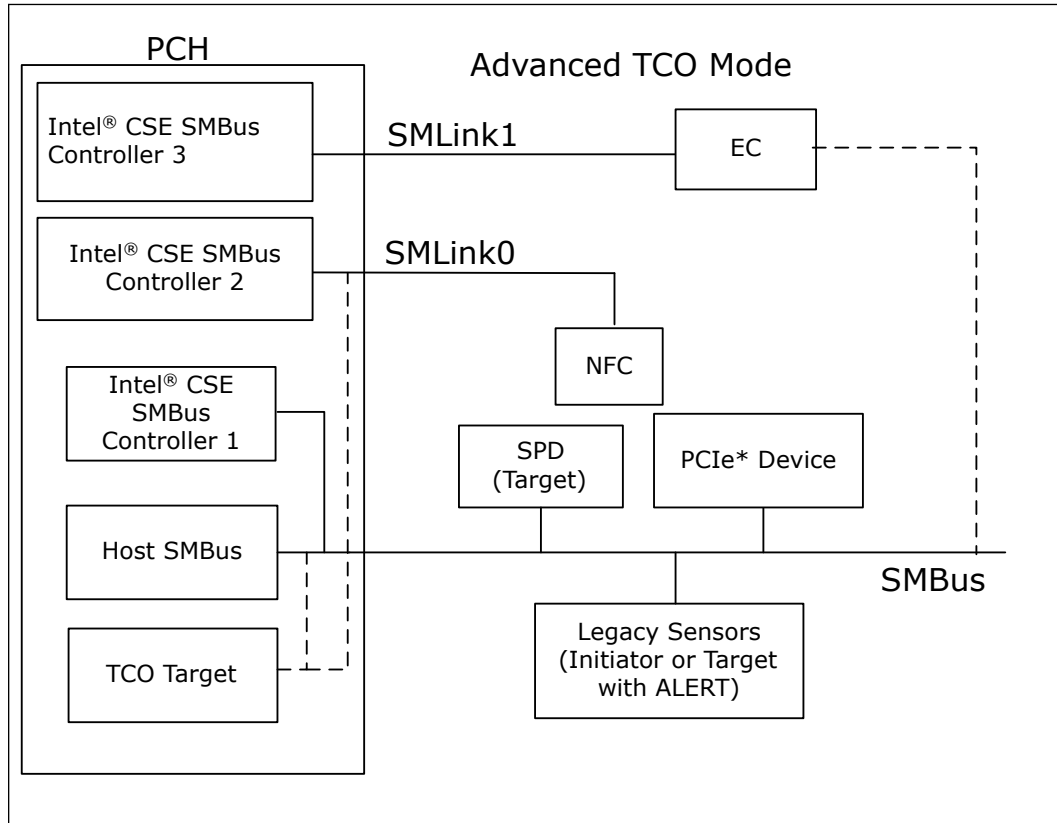
The PCH supports the Advanced TCO mode in which SMLink0 and SMLink1 are used in addition to the host SMBus.

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

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

SMLink1 can be connected to an Embedded Controller (EC).

Figure 28. Advanced TCO Mode



## 33.0 System Management Interface and SMLink

The PCH provides two SMLink interfaces, SMLink0 and SMLink1. The interfaces are intended for system management and are controlled by the Intel® CSE. Refer to [System Management](#) on page 241 for more detail.

**Table 105. Acronyms**

Acronyms	Description
EC	Embedded Controller

### 33.1 Functional Description

The SMLink interfaces are controlled by the Intel® CSE.

SMLink1 can be used with an Embedded Controller (EC).

Both SMLink0 and SMLink1 support up to 1 MHz.

### 33.2 Signal Description

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

*continued...*

Name	Type	Description
		External Pull-up resistor is required on this pin.
GPP_D14 / ISH_UART0_TXD / <b>SML0BCLK</b> / I2C6B_SCL	I/OD	System Management Link 0B Clock External Pull-up resistor is required on this pin.
GPP_D13 / ISH_UART0_RXD / <b>SML0BDATA</b> / I2C6B_SDA	I/OD	System Management Link 0B Data: SMBus link to external PHY. External Pull-up resistor is required on this pin.

### 33.3 Integrated Pull-Ups and Pull-Downs

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

### 33.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>INTRUDER#</b>	RTC	Undriven	Undriven	Undriven	OFF
<b>SML[1:0]DATA</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>SML[1:0]CLK</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>SML[1:0]ALERT#</b>	Primary	Pull-down (Internal)	Driven Low	Pull-down (Internal)	OFF
<b>PCHHOT#</b>	Primary	Pull-down (Internal)	Driven Low	Pull-down (Internal)	OFF
<b>PMCALERT#</b>	Primary	Undriven	Undriven	Undriven	OFF

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

## 34.0 Host System Management Bus (SMBus) Controller

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

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

**Table 106. Acronyms**

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

**Table 107. References**

Specification	Location
System Management Bus (SMBus) Specification, Version 2.0	<a href="http://www.smbus.org/specs/">http://www.smbus.org/specs/</a>

### 34.1 Functional Description

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

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

#### 34.1.1 Host Controller

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

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

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

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

### Host Controller Operation Overview

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

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

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

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

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

### Command Protocols

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

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

### Quick Command

When programmed for a Quick Command, the Transmit Target Address Register is sent. The PEC byte is never appended to the Quick Protocol. Software should force the PEC\_EN bit to 0 when performing the Quick Command. Software must force the I2C\_EN bit to 0 when running this command. Refer to Section 5.5.1 of the System Management Bus (SMBus) Specification, Version 2.0 for the format of the protocol.

### Send Byte/Receive Byte



For the Send Byte command, the Transmit Target Address and Device Command Registers are sent. For the Receive Byte command, the Transmit Target Address Register is sent. The data received is stored in the DATA0 register. Software must force the I2C\_EN bit to 0 when running this command.

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

### Write Byte/Word

The first byte of a Write Byte/Word access is the command code. The next 1 or 2 bytes are the data to be written. When programmed for a Write Byte/Word command, the Transmit Target Address, Device Command, and Data0 Registers are sent. In addition, the Data1 Register is sent on a Write Word command. Software must force the I2C\_EN bit to 0 when running this command. Refer to Section 5.5.4 of the System Management Bus (SMBus) Specification, Version 2.0 for the format of the protocol.

### Read Byte/Word

Reading data is slightly more complicated than writing data. First the PCH must write a command to the target device. Then it must follow that command with a repeated start condition to denote a read from that device's address. The target then returns 1 or 2 bytes of data. Software must force the I2C\_EN bit to 0 when running this command.

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

### Process Call

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

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

The Process Call command with I2C\_EN set and the PEC\_EN bit set produces undefined results. Software must force either I2C\_EN or PEC\_EN to 0 when running this command. Refer to Section 5.5.6 of the System Management Bus (SMBus) Specification, Version 2.0 for the format of the protocol.

---

### NOTES

1. For process call command, the value written into bit 0 of the Transmit Target Address Register needs to be 0.
  2. If the I2C\_EN bit is set, the protocol sequence changes slightly, the Command Code (Bits 18:11 in the bit sequence) are not sent. As a result, the target will not acknowledge (Bit 19 in the sequence).
- 

### Block Read/Write

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

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

For a Block Write, software must either force the I2C\_EN bit or both the PEC\_EN and AAC bits to 0 when running this command.

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

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

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

---

**NOTE**

For Block Write, if the I2C\_EN bit is set, the format of the command changes slightly. The PCH will still send the number of bytes (on writes) or receive the number of bytes (on reads) indicated in the DATA0 register. However, it will not send the contents of the DATA0 register as part of the message. When operating in I<sup>2</sup>C mode (I2C\_EN bit is set), the PCH will never use the 32 - byte buffer for any block commands.

---

**I<sup>2</sup>C\* Read**

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

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

---

**NOTE**

This command is supported independent of the setting of the I2C\_EN bit. The I<sup>2</sup>C Read command with the PEC\_EN bit set produces undefined results. Software must force both the PEC\_EN and AAC bit to 0 when running this command.

---

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

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

**Table 108. I<sup>2</sup>C\* Block Read**

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

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

**Block Write – Block Read Process Call**

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

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

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

- $M \geq 1$  byte
- $N \geq 1$  byte
- $M + N \leq 32$  bytes

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

---

**NOTES**

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

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

**Bus Arbitration**

Several initiators 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 initiator is driving the bus and the PCH will stop transferring data.

If the PCH detects that it has lost arbitration, the condition is called a collision. The PCH will set the BUS\_ERR bit in the Host Status Register, and if enabled, generates an interrupt or SMI#. The processor is responsible for restarting the transaction.

**Clock Stretching**

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

**Bus Timeout (PCH as SMBus Initiator)**

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

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

1. BYTE\_DONE\_STATUS bit (SMBus I/O Offset 00h, Bit 7) is set
2. The SECOND\_TO\_STS bit (TCO I/O Offset 06h, Bit 1) is not set (this indicates that the system has not locked up).

### Interrupts/SMI#

The PCH SMBus controller uses PIRQB# as its interrupt pin. However, the system can alternatively be set up to generate SMI# instead of an interrupt, by setting the SMBUS\_SMI\_EN bit.

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

**Table 109. Enable for SMBALERT#**

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

**Table 110. Enables for SMBus Target Write and SMBus Host Events**

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

**Table 111. Enables for the Host Notify Command**

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

### SMBus CRC Generation and Checking

If the AAC bit is set in the Auxiliary Control register, the PCH automatically calculates and drives CRC at the end of the transmitted packet for write cycles, and will check the CRC for read cycles. It will not transmit the contents of the PEC register for CRC. The PEC bit must not be set in the Host Control register if this bit is set, or unspecified behavior will result.

If the read cycle results in a CRC error, the DEV\_ERR bit and the CRCE bit in the Auxiliary Status register at Offset 0Ch will be set.

### 34.1.2 SMBus Target Interface

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

Key features and capabilities include:

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

---

#### NOTE

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

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

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

#### Format of Target Write Cycle

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

The table below has the values associated with the registers.

**Table 112. Target Write Registers**

Register	Function
0	Command Register. Refer to the table below for valid values written to this register.
1–3	Reserved
4	Data Message Byte 0
5	Data Message Byte 1
6–FFh	Reserved

*Note:* The external micro controller is responsible to make sure that it does not update the contents of the data byte registers until they have been read by the system processor. The PCH overwrites the old value with any new value received. A race condition is possible where the new value is being written to the register just at the time it is being read. The PCH will not attempt to cover this race condition (that is, unpredictable results in this case).

**Table 113. Command Types**

Command Type	Description
0	Reserved
1	<b>WAKE/SMI#.</b> This command wakes the system if it is not already awake. If system is already awake, an SMI# is generated.
2	<b>Unconditional Powerdown.</b> This command sets the PWRBTNOR_STS bit, and has the same effect as the Power button Override occurring.
3	<b>HARD RESET WITHOUT CYCLING:</b> This command causes a soft reset of the system (does not include cycling of the power supply). This is equivalent to a write to the CF9h register with Bits 2:1 set to 1, but Bit 3 set to 0.
4	<b>HARD RESET SYSTEM.</b> This command causes a hard reset of the system (including cycling of the power supply). This is equivalent to a write to the CF9h register with Bits 3:1 set to 1.
5	<b>Disable the TCO Messages.</b> This command will disable the PCH from sending Heartbeat and Event messages. Once this command has been executed, Heartbeat and Event message reporting can only be re-enabled by assertion and then de-assertion of the RSMRST# signal.
6	<b>WD RELOAD:</b> Reload watchdog timer.
7	Reserved
8	<b>SMLINK_SLV_SMI.</b> When the PCH detects this command type while in the S0 state, it sets the SMLINK_SLV_SMI_STS bit. This command should only be used if the system is in an S0 state. If the message is received during S3–S4 and S5 states, the PCH acknowledges it, but the SMLINK_SLV_SMI_STS bit does not get set. <i>Note:</i> It is possible that the system transitions out of the S0 state at the same time that the SMLINK_SLV_SMI command is received. In this case, the SMLINK_SLV_SMI_STS bit may get set but not serviced before the system goes to sleep. Once the system returns to S0, the SMI associated with this bit would then be generated. Software must be able to handle this scenario.
9–FFh	Reserved.

**Format of Read Command**

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

**Table 114. Target Read Cycle Format**

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

**Table 115. Data Values for Target Read Registers**

Register	Bits	Description
0	7:0	Reserved
1	2:0	<b>System Power State</b> 000 = S0 011 = S3 100 = S4 101 = S5 Others = Reserved
	7:3	Reserved
2	3:0	Reserved
	7:4	Reserved
3	5:0	<b>Watchdog Timer current value</b> <i>Note:</i> The Watchdog Timer has 10 bits, but this field is only 6 bits. If the current value is greater than 3Fh, the PCH will always report 3Fh in this field.
	7:6	Reserved
4	0	<b>Intruder Detect.</b> 1 = The Intruder Detect (INTRD_DET) bit is set. This indicates that the system cover has probably been opened.
	1	Reserved
	2	Reserved
	3	1 = <b>SECOND_TO_STS</b> bit set. This bit will be set after the second Timeout (SECOND_TO_STS bit) of the Watchdog Timer occurs.

*continued...*



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

- Behavioral Notes**

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

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

### Target Read of RTC Time Bytes

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

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

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

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

### Format of Host Notify Command

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

---

#### NOTE

Host software must always clear the HOST\_NOTIFY\_STS bit after completing any necessary reads of the address and data registers.

---

The table below shows the Host Notify format:

**Table 116. Host Notify Format**

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

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

### Format of Read Command

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

**Table 117. Target Read Cycle Format**

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

**Table 118. Data Values for Target Read Registers**

Register	Bits	Description
0	7:0	Reserved for capabilities indication. Should always return 00h. Future chips may return another value to indicate different capabilities.
1	2:0	<b>System Power State</b> 000 = S0 011 = S3 100 = S4 101 = S5

*continued...*

Register	Bits	Description
		Others = Reserved
	7:3	Reserved
2	3:0	Reserved
	7:4	Reserved
3	5:0	<b>Watchdog Timer current value</b> <i>Note:</i> The Watchdog Timer has 10 bits, but this field is only 6 bits. If the current value is greater than 3Fh, the PCH will always report 3Fh in this field.
	7:6	Reserved
4	0	<b>Intruder Detect.</b> 1 = The Intruder Detect (INTRD_DET) bit is set. This indicates that the system cover has probably been opened.
	1	<b>Temperature Event.</b> 1 = Temperature Event occurred. This bit will be set if the PCH's THRM# input signal is active. Else this bit will read "0."
	2	<b>DOA Processor Status.</b> This bit will be 1 to indicate that the processor is dead
	3	1 = <b>SECOND_TO_STS</b> bit set. This bit will be set after the second Timeout (SECOND_TO_STS bit) of the Watchdog Timer occurs.
	6:4	Reserved. Will always be 0, but software should ignore.
5	7	<b>SMBALERT# Status:</b> Reflects the value of the GPIO11/SMBALERT# pin (when the pin is configured as SMBALERT#). Valid only if SMBALERT_DISABLE = 0. Value always return 1 if SMBALERT_DISABLE = 1. (high = 1, low = 0).
	0	<b>FWH bad bit:</b> This bit will be 1 to indicate that the FWH read returned FFh, which indicates that it is probably blank.
	1	<b>Battery Low Status:</b> 1 if the BATLOW# pin is a 0.
	2	<b>SYS_PWROK Failure Status:</b> This bit will be 1 if the SYSPWR_FLR bit in the GEN_PMCON_2 register is set.
	3	Reserved
	4	Reserved
	5	<b>POWER_OK_BAD:</b> Indicates the failure core power well ramp during boot/resume. This bit will be active if the SLP_S3# pin is de-asserted and PCH_PWROK pin is not asserted.
	6	<b>Thermal Trip:</b> This bit will shadow the state of processor Thermal Trip status bit (CTS). Events on signal will not create a event message.
7	Reserved: Default value is "X" <i>Note:</i> Software should not expect a consistent value when this bit is read through SMBUS/SMLink	
6	7:0	Contents of the Message 1 register.
7	7:0	Contents of the Message 2 register.
8	7:0	Contents of the WDSTATUS register.
9	7:0	Seconds of the RTC
A	7:0	Minutes of the RTC
B	7:0	Hours of the RTC
C	7:0	"Day of Week" of the RTC
D	7:0	"Day of Month" of the RTC
<b>continued...</b>		

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

**Table 119. Enables for SMBus Target Write and SMBus Host Events**

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

## 34.2 SMBus Power Gating

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

## 34.3 Signal Description

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

## 34.4 Integrated Pull-Ups and Pull-Downs

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

## 34.5 I/O Signal Planes and States

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

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

## 35.0 Serial Peripheral Interface (SPI)

The PCH provides two Serial Peripheral Interfaces (SPI). The SPI0 interface consists of three Chip Select signals. SPI0 interface can allow two flash memory devices (SPI0\_CS0# and SPI0\_CS1#) and one TPM device (SPI0\_CS2#) to be connected to the PCH. The SPI0 interface support either 1.8 V or 3.3 V. The voltage is selected via a Hardware strap on SPIVCCIOSEL signal. Refer to [VCCSPI Voltage \(3.3 V or 1.8 V\) Selection](#) on page 269.

**Table 120. Acronyms**

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

### 35.1 Functional Description

This section provides information on the following topics:

- SPI0 for Flash
- SPI0 support for TPM

#### 35.1.1 SPI0 for Flash

The Serial Peripheral Interface (SPI0) supports two SPI flash devices via two chip select (SPI0\_CS0# and SPI0\_CS1#). The maximum size of flash supported is determined by the SFDP-discovered addressing capability of each device. Each component can be up to 16 MB (32 MB total addressable) using 3-byte addressing. Each component can be up to 64 MB (128 MB total addressable) using 4-byte addressing. Another chip select (SPI0\_CS2#) is also available and only used for TPM on SPI support. PCH drives the SPI0 interface clock at either 14 MHz, 25 MHz, 33 MHz, or 50 MHz and will function with SPI flash/TPM devices that support at least one of these frequencies. The SPI interface supports either 3.3 V or 1.8 V.

A SPI0 flash device supporting SFDP (Serial Flash Discovery Parameter) is required for all PCH design. A SPI0 flash device on SPI0\_CS0# with a valid descriptor MUST be attached directly to the PCH.

The PCH supports fast read which consist of:

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

The PCH SPI0 has a third chip select SPI0\_CS2# for TPM support over SPI. The TPM on SPI0 will use SPI0\_CLK, SPI0\_MISO, SPI0\_MOSI and SPI0\_CS2# SPI signals.

### SPI0 Supported Features

- **Descriptor Mode**  
Descriptor Mode is required for all SKUs of the PCH. Non-Descriptor Mode is not supported.
- **SPI0 Flash Regions**  
In Descriptor Mode the Flash is divided into five separate regions.

**Table 121. SPI0 Flash Regions**

Region	Content
0	Flash Descriptor
1	BIOS
2	Intel Converged Security Engine
4	Platform Data
5	EC

Only three initiators can access the regions: Host processor running BIOS code, Intel Converged Security Engine, and the EC.

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

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

### Flash Region Sizes

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



**Table 122. Region Size Versus Erase Granularity of Flash Components**

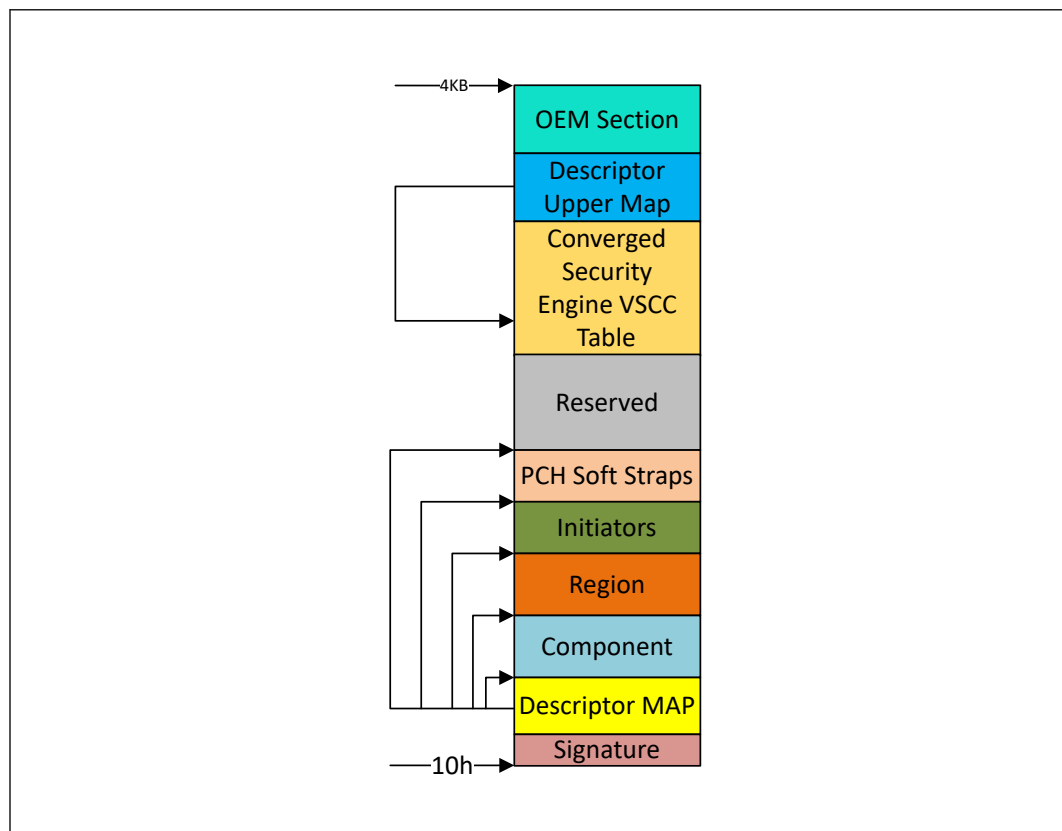
Region	Size with 4 KB Blocks	Size with 8 KB Blocks	Size with 64 KB Blocks
Descriptor	4 KB	8 KB	64 KB
BIOS	Varies by Platform	Varies by Platform	Varies by Platform
Intel® CSE	Varies by Platform	Varies by Platform	Varies by Platform
EC	Varies by Platform	Varies by Platform	Varies by Platform

**Flash Descriptor**

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

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

**Figure 29. Flash Descriptor Regions**



- The Flash signature selects Descriptor Mode as well as verifies if the flash is programmed and functioning. The data at the bottom of the flash (offset 10h) must be 0FF0A55Ah in order to be in Descriptor mode.

- The Descriptor map has pointers to the other five descriptor sections as well as the size of each.
- The component section has information about the SPI0 flash in the system including: the number of components, density of each, invalid instructions (such as chip erase), and frequencies for read, fast read and write/erase instructions.
- The Region section points to the three other regions as well as the size of each region.
- The Initiator region contains the security settings for the flash, granting read/write permissions for each region and identifying each initiator by a requester ID.
- The processor and PCH Soft Strap sections contain processor and PCH configurable parameters.
- The Reserved region between the top of the processor strap section and the bottom of the OEM Section is reserved for future chipset usages.
- The Descriptor Upper MAP determines the length and base address of the Management Engine VSCC Table.
- The Management Engine VSCC Table holds the JEDEC ID and the VSCC information of the entire SPI0 Flash supported by the NVM image.
- OEM Section is 256 bytes reserved at the top of the Flash Descriptor for use by OEM.

- **Descriptor Initiator Region**

The initiator region defines read and write access setting for each region of the SPI0 device. The initiator region recognizes three initiators: BIOS, CSE, and EC. Each initiator is only allowed to do direct reads of its primary regions.

**Table 123. Region Access Control Table**

Initiator Read/Write Access			
Region	Processor and BIOS	Intel® CSE	EC
BIOS	Read/Write	N/A	Note
Intel® Converged Security Engine (CSE)	N/A	Read/Write	N/A
EC	Read	N/A	Read/Write

*Note:* Optional BIOS access to the EC region.

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

Region Name	Starting Address
Signature	10h
Component FCBA	30h
Regions FRBA	40h
Initiators FMBA	80h
PCH Straps FPSBA	100h
MDTBA	C00h
PMC Straps	C14h

**continued...**

Region Name	Starting Address
CPU Straps	C2Ch
Intel® CSE Straps	C3Ch
Register Init FIBA	340h

### Flash Access

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

- **Direct Access**

- Initiators are allowed to do direct read only of their primary region
- Initiator's Host or Management Engine virtual read address is converted into the SPI0 Flash Linear Address (FLA) using the Flash Descriptor Region Base/Limit registers

#### Direct Access Security

- Requester ID of the device must match that of the primary Requester ID in the Initiator 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 initiator

- **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 initiators can access the registers

## 35.1.2 SPI0 support for TPM

The PCH's SPI0 flash controller supports a discrete TPM on the platform via its dedicated SPI0\_CS2# signal. The platform must have no more than 1 TPM.

SPI0 controller supports accesses to SPI0 TPM at approximately 14 MHz, 33 MHz and 50 MHz depending on the PCH soft strap. 20 MHz is the reset default, a valid PCH soft strap setting overrides the requirement for the 20 MHz. SPI0 TPM device must support a clock of 20 MHz, and thus should handle 15-20 MHz. It may but is not required to support a frequency greater than 20 MHz.

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

## 35.2 Signal Description

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

## 35.3 Integrated Pull-Ups and Pull-Downs

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

### NOTE

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

## 35.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>SPI0_CLK</b>	Primary	Internal Pull-down	Driven Low	Driven Low	OFF
<b>SPI0_MOSI</b>	Primary	Hi-Z	Internal PU, then Driven Low	Driven Low	OFF

*continued...*

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
		(Refer to Note 2)			
<b>SPIO_MISO</b>	Primary	Hi-Z	Internal Pull-up	Internal Pull-up	OFF
<b>SPIO_CS0#</b>	Primary	Internal Pull-down	Driven High	Driven High	OFF
<b>SPIO_CS1#</b>	Primary	Internal Pull-down	Driven High	Driven High	OFF
<b>SPIO_CS2#</b>	Primary	Internal Pull-down	Driven High	Driven High	OFF
<b>SPIO_IO[3:2]</b>	Primary	Hi-Z (Refer to Note 2)	Internal Pull-up	Internal Pull-up	OFF

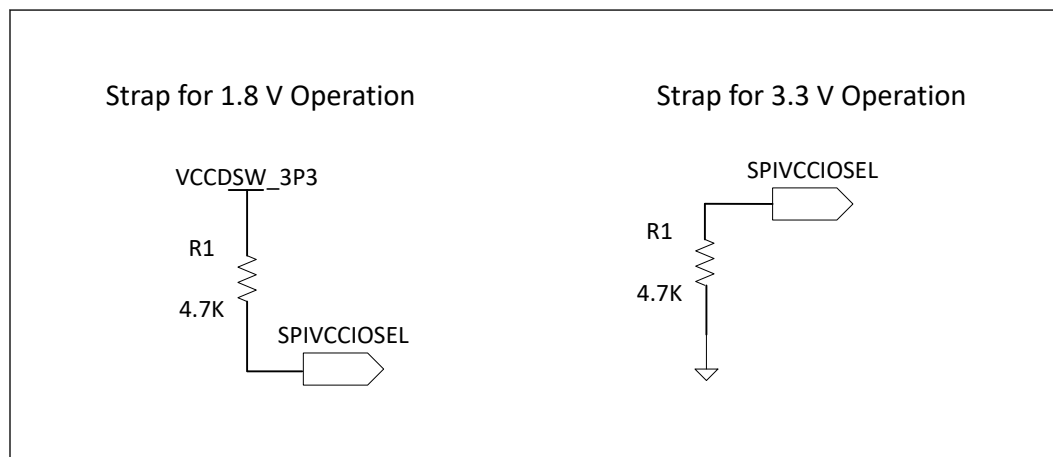
Notes: 1. During reset refers to when RSMRST# is asserted.  
 2. SPIO\_MOSI, SPIO\_IO[3:2] also function as strap pins. The actual pin state during Reset is dependent on the platform Pull-up/Pull-down resistor.

### 35.5 VCCSPI Voltage (3.3 V or 1.8 V) Selection

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

- 0 = SPI voltage is 3.3 V (4.7 kohm pull-down to GND)
- 1 = SPI voltage is 1.8 V (4.7 kohm pull-up to VCCDSW\_3P3)

Figure 30. VCCSPI Voltage (3.3 V or 1.8 V) Selection



## 36.0 Enhanced Serial Peripheral Interface (eSPI)

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

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

**Table 124. Acronyms**

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

**Table 125. References**

Specification	Document Number/Location
Enhanced Serial Peripheral Interface (eSPI) Specifications	<a href="https://downloadcenter.intel.com/download/27055/eSPI">https://downloadcenter.intel.com/download/27055/eSPI</a>

### 36.1 Functional Description

This section provides information on the following topics:

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

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

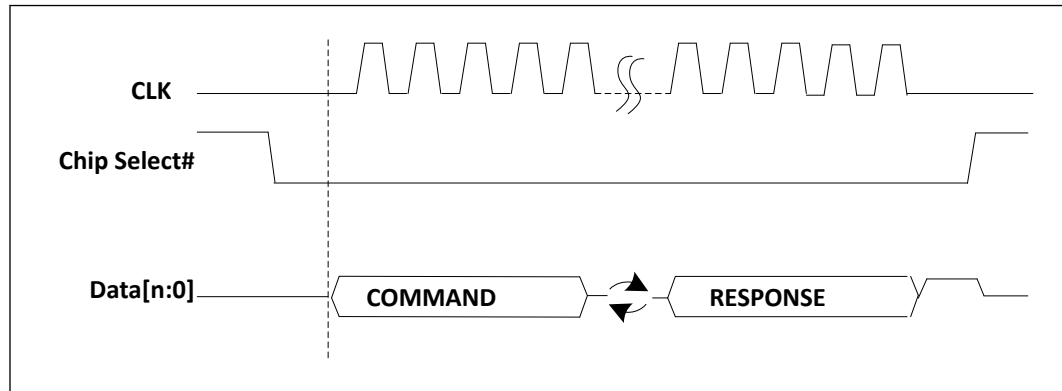
### 36.1.1 Operating Frequency

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

### 36.1.2 Protocols

Below is an overview of the basic eSPI protocol. Refer to the latest eSPI Specification and corresponding platform eSPI Compatibility Specification for more details (Refer to [Table 125](#) on page 270).

**Figure 31. Basic eSPI Protocol**



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

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 target.

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

### 36.1.3 WAIT States from eSPI Target

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

### 36.1.4 In-Band Link Reset

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

### 36.1.5 Target Discovery

The PCH eSPI interface is enabled using a hard pin strap. Refer to [Pin Straps](#) on page 44 for details on the strap.

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

### 36.1.6 Flash Sharing Mode

eSPI supports Initiator Attached Flash sharing (abbreviated in this as MAFS). The Flash sharing mode selected for a specific platform is dependent on strap settings.

### 36.1.7 PECI Over eSPI

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

The PECI bus may be connected to the PCH via either the legacy PECI pin or the eSPI interface. The operation via legacy PECI pin or over eSPI is selected via a soft strap and only one or the other is enabled in a given platform.

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

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

### 36.1.8 Multiple OOB Initiator

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

### 36.1.9 Channels and Supported Transactions

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



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

The number of types of channels supported by a particular eSPI target is discovered through the GET\_CONFIGURATION command issued by the PCH to the eSPI target during initialization.

Table below summarizes the eSPI channels and supported transactions.

**Table 126. eSPI Channels and Supported Transactions**

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

### Peripheral Channel (Channel 0) Overview

The Peripheral channel performs the following functions:

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

### Virtual Wire Channel (Channel 1) Overview

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

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

Table below summarizes the PCH virtual wires in eSPI mode.

**Table 127. eSPI Virtual Wires (VW)**

Virtual Wire	PCH Pin Direction	Reset Control	Pin Retained in PCH (For Use by Other Components)
SUS_STAT#	Output	ESPI_RESET#	No
SUSWARN#	Output	ESPI_RESET#	No
<i>continued...</i>			

Virtual Wire	PCH Pin Direction	Reset Control	Pin Retained in PCH (For Use by Other Components)
SUS_ACK	Input	ESPI_RESET#	No
SUSPWRDNACK	Output	ESPI_RESET#	No
PLTRST#	Output	ESPI_RESET#	Yes
PME# (eSPI Peripheral PME)	Input	ESPI_RESET#	N/A
WAKE#	Input	ESPI_RESET#	No
SMI#	Input	PLTRST#	N/A
SCI#	Input	PLTRST#	N/A
RCIN#	Input	PLTRST#	No
SLP_A#	Output	ESPI_RESET#	Yes
SLP_S4#/SLP_S5#/ SLP_WLAN#	Output	DSW_PWROK	Yes
TARGET_BOOT_LOAD_DONE	Input	ESPI_RESET#	N/A
TARGET_BOOT_LOAD_STAT US	Input	ESPI_RESET#	N/A
HOST_RST_WARN	Output	PLTRST#	N/A
HOST_RST_ACK	Input	PLTRST#	N/A
OOB_RST_WARN	Output	ESPI_RESET#	N/A
OOB_RST_ACK	Input	ESPI_RESET#	N/A
HOST_C10	Output	PLTRST#	N/A
ERROR_NONFATAL	Input	ESPI_RESET#	N/A
ERROR_FATAL	Input	ESPI_RESET#	N/A

**Interrupt Events**

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

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

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

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

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

- Tunnel MCTP Packets between the Intel® CSE and eSPI Target Device:** The Intel® CSE 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 target 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 CSE and eSPI device.

- **Tunnel PCH Temperature Data to the eSPI Target:** The eSPI controller stores the PCH temperature data internally and sends it to the target 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 Target:** the eSPI controller captures this data internally at periodic intervals from the PCH RTC controller and sends it to the target device using a posted OOB message when a request is made to a specific destination address.
- **PCH Temperature Data Over eSPI OOB Channel**

eSPI controller supports the transmitting of PCH thermal data to the eSPI target. 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 target and the PCH response back are shown in the two figures below.

**Figure 32. eSPI Target 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 Target Addr. = 01h (PCH OOB HW Handler)							0
4	Common code = 01h (Get_PCH_Temp)							
5	Byte Count = 01h							
6	Source Target Address[7:0] = 0Fh (eSPI Target 0/EC)							1

**Figure 33. PCH Response to eSPI Target 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 Target Addr. = 0Eh (eSPI Target 0/EC)							0
4	Common code = 01h (Get_PCH_Temp)							
5	Byte Count = 02h							
6	Source Target Address [7:0] = 01h (PCH OOB HW Handler)							1
7	PCH Temperature Data [7:0]							

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

The PCH eSPI controller supports the transmitting of PCH RTC time/date to the eSPI target. This allows the eSPI target 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 target and the PCH response back to the device are shown in the two figures below.

**Figure 34. eSPI Target Request to PCH for PCH RTC Time**

Byte #	7	6	5	4	3	2	1	0
0	eSPI Cycle Type: OOB Message = 21h							
1	Tag[3:0]			Length[11:8] = 0h				
2	Length[7:0] = 04h							
3	Dest Target Addr. [7:1] = 01h (PCH OOB HW Handler)							0
4	Common code = 02h (Get_PCH_RTC_Time)							
5	Byte Count = 01h							
6	Source Target Address [7:0] = 0Fh (eSPI Target 0/EC)							1

Figure 35. PCH Response to eSPI Target with RTC Time

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

**NOTES**

- 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.
- 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.
- DM:** Data Mode. A 1 indicates that the time byte are specified in binary. A 0 indicates that the time bytes are in the Binary Coded Decimal (BCD) format.

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

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

The Initiator Attached Flash Channel controller (MAFCC) tunnels flash accesses from eSPI target 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 target in a separate completion packet.

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

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

## 36.2 Signal Description

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

## 36.3 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
<b>ESPI_IO[3:0]</b>	Pull-up	20 kohm +/- 30%	
<b>ESPI_CLK</b>	Pull-down	20 kohm +/- 30%	
<b>ESPI_CS [1:0]#</b>	Pull-up	20 kohm +/- 30%	
<b>ESPI_ALERT#</b>	Pull-up	15 - 40 kohm	

## 36.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>ESPI_IO [3:0]</b>	Primary	Internal Pull-up	Internal Pull-up	Internal Pull-up	OFF
<b>ESPI_CLK</b>	Primary	Internal Pull-down	Driven Low	Driven Low	OFF
<b>ESPI_CS [1:0] #</b>	Primary	Internal Pull-up	Driven High	Driven High	OFF
<b>ESPI_ALERT#</b>	Primary	Internal Pull-up	Driven High	Driven High	OFF
<b>ESPI_RESET#</b>	Primary	Driven Low	Driven High	Driven High	OFF
<i>Note:</i> Reset reference for primary well pins is RSMRST#.					

## 37.0 Intel® Serial IO Generic SPI (GSPI) Controllers

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

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

The GSPI interfaces support the following features:

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

---

### NOTE

Target mode is not supported.

---

**Table 128. Acronyms**

Acronyms	Description
GSPI	Generic Serial Peripheral Interface
LTR	Latency Tolerance Reporting

## 37.1 Functional Description

This section provides information on the following topics:

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

### 37.1.1 Controller Overview

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

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



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

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

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

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

The GSPI controller, as a initiator, 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.

### 37.1.2 DMA Controller

The GSPI controllers have an integrated DMA controller.

#### DMA Transfer and Setup Modes

The DMA can operate in the following modes:

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

The DMA supports the following modes for programming:

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

#### Channel Control

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

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

### 37.1.3 Reset

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

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

### 37.1.4 Power Management

#### Device Power Down Support

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

#### Latency Tolerance Reporting (LTR)

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

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

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

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

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

### 37.1.5 Interrupts

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

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

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

### 37.1.6 Error Handling

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

## 37.2 Signal Description

Name	Type	Description
GPP_E10 / THC0_SPI1_CS# / <b>GSPI0_CS0#</b>	O	<b>Generic SPI 0 Chip Select</b>
GPP_E11 / THC0_SPI1_CLK / <b>GSPI0_CLK</b>	O	<b>Generic SPI 0 Clock</b>
GPP_E12 / THC0_SPI1_IO1 / I2C0A_SDA / <b>GSPI0_MISO</b>	I	<b>Generic SPI 0 MISO</b>
GPP_E13 / THC0_SPI1_IO0 / I2C0A_SCL / <b>GSPI0_MOSI</b>	O	<b>Generic SPI 0 MOSI</b> <i>Note: This signal is also utilized as a strap. Refer to the pin strap section for more information.</i>
GPP_F16 / GSXCLK / THC1_SPI2_CS# / <b>GSPI1_CS0#</b>	O	<b>Generic SPI 1 Chip Select 0</b>
GPP_F11 / THC1_SPI2_CLK / <b>GSPI1_CLK</b>	O	<b>Generic SPI 1 Clock</b>
GPP_F13 / GSXSLOAD / THC1_SPI2_IO1 / <b>GSPI1_MISO</b>	I	<b>Generic SPI 1 MISO</b>
GPP_F12 / GSXDOUT / THC1_SPI2_IO0 / <b>GSPI1_MOSI</b>	O	<b>Generic SPI 1 MOSI</b> <i>Note: This signal is also utilized as a strap. Refer to the pin strap section for more information.</i>

*continued...*

Name	Type	Description
GPP_D9 / ISH_SPI_CS# / BSSB_LS2_RX / <b>GSPI2_CS0#</b>	0	<b>Generic SPI 2 Chip Select 0</b>
GPP_D10 / ISH_SPI_CLK / BSSB_LS2_TX / <b>GSPI2_CLK</b>	0	<b>Generic SPI 2 Clock</b>
GPP_D11 / ISH_SPI_MISO / BSSB_LS3_RX / <b>GSPI2_MISO</b>	I	<b>Generic SPI 2 MISO</b>
GPP_D12 / ISH_SPI_MOSI / BSSB_LS3_TX / <b>GSPI2_MOSI</b>	0	<b>Generic SPI 2 MOSI</b>

### 37.3 Integrated Pull-Ups and Pull-Downs

Signal	Resistor Type	Value	Notes
<b>GSPI0_MOSI</b>	Pull Down	20 kohm ± 30%	The integrated pull down is disabled within 2 RTC clocks after PCH_PWROK assertion
<b>GSPI1_MOSI</b>	Pull Down	20 kohm ± 30%	The integrated pull down is disabled within 2 RTC clocks after PCH_PWROK assertion
<b>GSPI2_MOSI</b>	Pull Down	20 kohm ± 30%	The integrated pull down is disabled within 2 RTC clocks after PCH_PWROK assertion
<b>GSPI0_MISO</b>	Pull Down	20 kohm ± 30%	
<b>GSPI1_MISO</b>	Pull Down	20 kohm ± 30%	
<b>GSPI2_MISO</b>	Pull Down	20 kohm ± 30%	

### 37.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>GSPI0_CS0#, GSPI1_CS0#, GSPI2_CS0#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>GSPI2_CLK, GSPI1_CLK, GSPI0_CLK</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>GSPI2_MISO, GSPI1_MISO, GSPI0_MISO</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>GSPI2_MOSI, GSPI1_MOSI, GSPI0_MOSI</b>	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF

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

## 38.0 Touch Host Controller (THC)

Touch Host Controller provides low latency interface for internal HID peripherals (such as touch screen and touch pad). In the first generation THC, only SPI IOs (single/dual/quad) are supported.

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

**Table 129. Acronyms**

Acronyms	Description
CLK	Clock
CS	Chip Select
MISO	Initiator In Target Out
MOSI	Initiator Out Target In
TPM	Trusted Platform Module

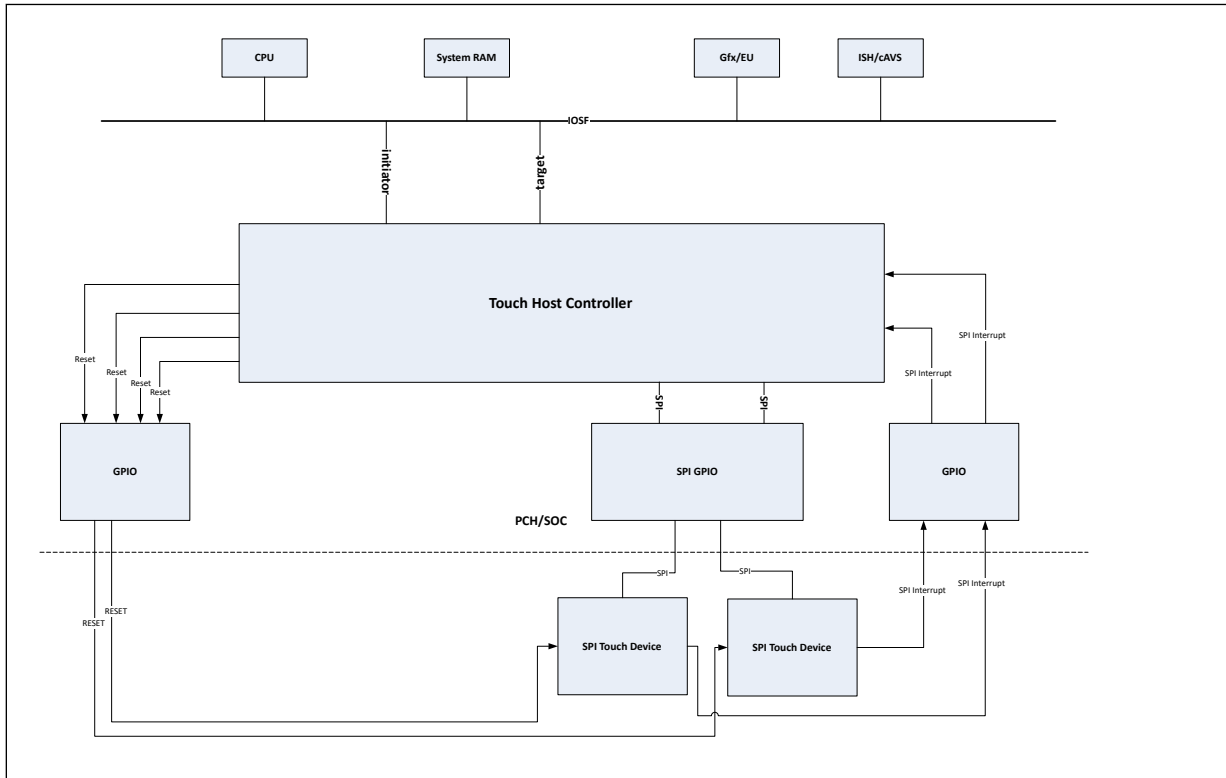
### 38.1 Functional Description

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

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

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

Figure 36. THC Block Diagram



### 38.2 Signal Description

Name	Type	Description
GPP_E11 / <b>THC0_SPI1_CLK</b> / GSPI0_CLK	O	<b>THC0_SPI1 Clock:</b> THC SPI1 clock output from PCH. Supports 18MHz, 21MHz, 26MHz, 32MHz, and 42MHz.
GPP_F11 / <b>THC1_SPI2_CLK</b> / GSPI1_CLK	O	<b>THC1_SPI2 Clock:</b> THC SPI2 clock output from PCH.
GPP_E10 / <b>THC0_SPI1_CS#</b> / GSPI0_CS0#	O	<b>THC0_SPI1 Chip Select:</b> Used to select the touch devices if it is connected to THC0_SPI1 interface.
GPP_F16 / GSXCLK / <b>THC1_SPI2_CS#</b> / GSP1_CS0#	O	<b>THC1_SPI2 Chip Select:</b> Used to select the touch devices if it is connected to THC1_SPI2 interface.
GPP_E13 / <b>THC0_SPI1_IO0</b> / GSPI0_MOSI	I/O	<b>THC0_SPI1_IO0:</b> A bidirectional signal used to support single, dual and quad mode data transfer.
GPP_E12 / <b>THC0_SPI1_IO1</b> / GSPI0_MISO	I/O	<b>THC0_SPI1_IO1:</b> A bidirectional signal used to support single, dual and quad mode data transfer.
GPP_E1 / <b>THC0_SPI1_IO2</b>	I/O	<b>THC0_SPI1_IO2:</b> A bidirectional signal used to support single, dual and quad mode data transfer.

*continued...*

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

### 38.3 Integrated Pull-Ups and Pull-Downs

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

#### NOTE

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

### 38.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
THC0_SPI1_CLK	Primary	Undriven	Undriven	Undriven	OFF
THC1_SPI2_CLK	Primary	Undriven	Undriven	Undriven	OFF
THC0_SPI1_CS#	Primary	Undriven	Undriven	Undriven	OFF
THC1_SPI2_CS#	Primary	Undriven	Undriven	Undriven	OFF
THC0_SPI1_IO[0:3]	Primary	Undriven	Undriven	Undriven	OFF
THC1_SPI2_IO[0:3]	Primary	Undriven	Undriven	Undriven	OFF
THC0_SPI1_RST#	Primary	Undriven	Undriven	Undriven	OFF
THC1_SPI2_RST#	Primary	Undriven	Undriven	Undriven	OFF

*continued...*



Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
THC0_SPI1_INT#	Primary	Undriven	Undriven	Undriven	OFF
THC1_SPI2_INT#	Primary	Undriven	Undriven	Undriven	OFF

Note: 1. During reset refers to when RSMRST# is asserted.



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

---

The PCH implements three independent UART interfaces, UART0, UART1 and UART2. Each UART interface is a 4-wire interface supporting up to 6.25 Mbit/s.

The interfaces can be used in the low-speed, full-speed, and high-speed modes. The UART communicates with serial data ports that conform to the RS-232 interface protocol.

UART2 only implements the UART Host controller and does not incorporate a DMA controller which is implemented for UART0 and UART1. Therefore, UART2 is restricted to operate in PIO mode only.

The UART interfaces support the following features:

- Up to 6.25 Mbit/s Auto Flow Control mode as specified in the 16750 standard
- Transmitter Holding Register Empty (THRE) interrupt mode
- 64-byte TX and 64-byte RX host controller FIFOs
- DMA support with 64-byte DMA FIFO per channel (up to 32-byte burst)
- Functionality based on the 16550 industry standards
- Programmable character properties, such as number of data bits per character (5-8), optional parity bit (with odd or even select) and number of stop bits (1, 1.5, or 2)
- Line break generation and detection
- DMA signaling with two programmable modes
- Prioritized interrupt identification
- Programmable FIFO enable/disable
- Programmable serial data baud rate
- Modem and status lines are independently controlled
- Programmable BAUD RATE supported (baud rate = (serial clock frequency)/(16xdivisor))

---

### NOTES

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

**Table 130. Acronyms**

Acronyms	Description
DMA	Direct Memory Access
UART	Universal Asynchronous Receiver/Transmitter
LSx	Low speed IO Controller

## 39.1 Functional Description

This section provides information on the following topics:

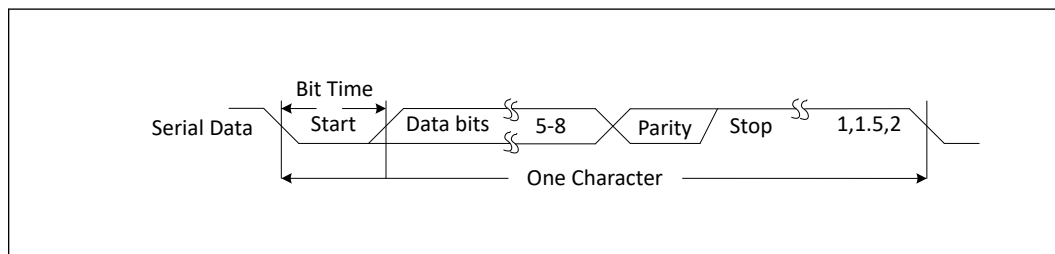
- UART Serial (RS-232) Protocols Overview
- 16550 8-bit Addressing - Debug Driver Compatibility
- DMA Controller
- Reset
- Power Management
- Interrupts
- Error Handling

### 39.1.1 UART Serial (RS-232) Protocols Overview

Because the serial communication between the UART host controller and the selected device is asynchronous, Start and Stop bits are used on the serial data to synchronize the two devices. The structure of serial data accompanied by Start and Stop bits is referred to as a character.

An additional parity bit may be added to the serial character. This bit appears after the last data bit and before the stop bit(s) in the character structure to provide the UART Host Controller with the ability to perform simple error checking on the received data.

**Figure 37. UART Serial Protocol**



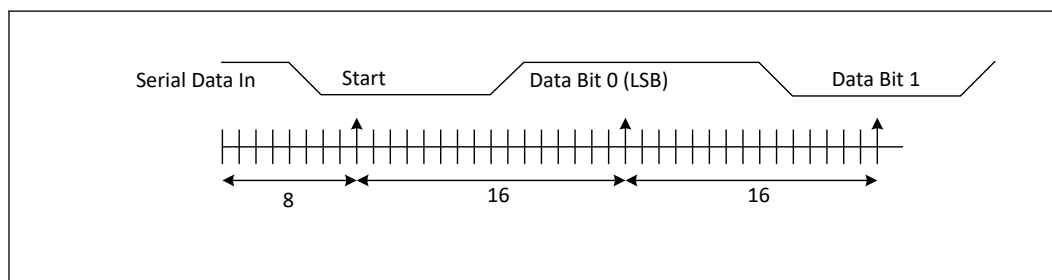
The UART Host Controller Line Control Register (LCR) is used to control the serial character characteristics. The individual bits of the data word are sent after the Start bit, starting with the least significant bit (LSB). These are followed by the optional parity bit, followed by the Stop bit(s), which 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 38. UART Receiver Serial Data Sample Points**



### 39.1.2 16550 8-bit Addressing - Debug Driver Compatibility

#### NOTE

The PCH UART host controller is not compatible with legacy UART 16550 debug-port drivers. The UART host controller operates in 32-bit addressing mode only. UART 16550 legacy drivers only operate with 8-bit addressing. In order to provide compatibility with standard in-box legacy UART drivers a 16550 Legacy Driver mode has been implemented in the UART controller that will convert 8-bit addressed accesses from the 16550 legacy driver to the 32-bit addressing that the UART host controller supports. The UART 16550 8-bit Legacy mode only operates with PIO transactions. DMA transactions are not supported in this mode.

### 39.1.3 DMA Controller

The UART controllers 0 and 1 (UART0 and UART1) have an integrated DMA controller. Each channel contains a 64-byte FIFO. Max. burst size supported is 32 bytes.

UART controller 2 (UART2) only implements the host controllers and does not incorporate a DMA. Therefore, UART2 is restricted to operate in PIO mode only.

#### DMA Transfer and Setup Modes

The DMA can operate in the following modes:

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

The DMA supports the following modes for programming:

1. Direct programming. Direct register writes to DMA registers to configure and initiate the transfer.

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

#### Channel Control

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

### 39.1.4 Reset

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

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

### 39.1.5 Power Management

#### Device Power Down Support

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

#### Latency Tolerance Reporting (LTR)

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

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

1. Platform/HW Default Control. This scheme is used for usage models in which the controller’s state correctly informs the platform of the current latency requirements. In this scheme, the latency requirement is a function of the controller state. The latency for transmitting data to/from its connected device at a given rate while the controller is active is representative of the active latency requirements. On the other hand if the device is not transmitting or receiving data and idle, there is no expectation for end to end latency.
2. Driver Control. This scheme is used for usage models in which the controller state does not inform the platform correctly of the current latency requirements. If the FIFOs of the connected device are much smaller than the controller FIFOs, or the connected device’s end to end traffic assumptions are much smaller than the latency to restore the platform from low power state, driver control should be used.

### 39.1.6 Interrupts

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

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

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

### 39.1.7 Error Handling

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

## 39.2 Signal Description

Name	Type	Description
GPP_H10 / <b>UART0_RXD</b> / M2_SKT2_CFG0	I	<b>UART 0 Receive Data</b>
GPP_H11 / <b>UART0_TXD</b> / M2_SKT2_CFG1	O	<b>UART 0 Transmit Data</b>
GPP_H12 / I2C7_SDA/ <b>UART0_RTS#</b> / M2_SKT2_CFG0 / ISH_GP6B / DEVSLP0B#	O	<b>UART 0 Request to Send</b>
GPP_H13 / I2C7_SCL / <b>UART0_CTS#</b> / M2_SKT2_CFG1 / ISH_GP7B / DEVSLP1B#	I	<b>UART 0 Clear to Send</b>
<i>continued...</i>		

Name	Type	Description
GPP_D17 / <b>UART1_RXD</b> / ISH_UART1_RXD	I	<b>UART 1 Receive Data</b>
GPP_D18 / <b>UART1_TXD</b> / ISH_UART1_TXD	O	<b>UART 1 Transmit Data</b>
GPP_F1 / CNV_BRI_RSP / <b>UART2_RXD</b>	I	<b>UART 2 Receive Data</b>
GPP_F2 / CNV_RGI_DT / <b>UART2_TXD</b>	O	<b>UART 2 Transmit Data</b>
GPP_F0 / CNV_BRI_DT / <b>UART2_RTS#</b>	O	<b>UART 2 Request to Send</b>
GPP_F3 / CNV_RGI_RSP / <b>UART2_CTS#</b>	I	<b>UART 2 Clear to Send</b>

### 39.3 Integrated Pull-Ups and Pull-Downs

None.

### 39.4 I/O Signal Planes and States

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>UART[2:0]_RXD</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>UART[2:0]_TXD</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>UART[2:0]_RTS#</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>UART[2:0]_CTS#</b>	Primary	Undriven	Undriven	Undriven	OFF

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

## 40.0 Private Configuration Space Target Port ID

The PCH incorporates a wide variety of devices and functions. The registers within these devices are mainly accessed through the primary interface, such as PCI configuration space and IO/MMIO space. Some devices also have registers that are distributed within the PCH Private Configuration Space at individual endpoints (Target Port IDs) which are only accessible through the PCH Sideband Interface. These PCH Private Configuration Space Registers can be addressed via SBREG\_BAR or through SBI Index Data pair programming.

**Table 131. Private Configuration Space Register Target Port IDs**

PCH Device/Function Type	Target Port ID
OPI Configuration	88h
FIA Configuration	CFh
General Purpose I/O (GPIO) Community 0	6Eh
General Purpose I/O (GPIO) Community 1	6Dh
General Purpose I/O (GPIO) Community 2	6Ch
General Purpose I/O (GPIO) Community 4	6Ah
DCI	71h
PCIe Controller #1 (SPA)	80h
PCIe Controller #2 (SPB)	81h
PCIe Controller #3 (SPC)	82h
SATA	D9h
SMBus	C6h
eSPI / SPI	72h
xHCI	70h
CNVi	73h
HSIO Strap Configuration	89h
Real Time Clock (RTC)	C3h
Processor Interface, 8254 Timer, HPET, APIC	C4h
USB 2.0	CAh
UART, I <sup>2</sup> C, GSPI	CBh
Integrated Clock Controller (ICC)	DCh
eMMC	A1h
General Purpose I/O (GPIO) Community 5	69h
USB Dual Role / OTG	E5h
<i>continued...</i>	



<b>PCH Device/Function Type</b>	<b>Target Port ID</b>
MODPHY0	ABh
MODPHY1	AAh
MODPHY2	A9h
MODPHY3	A8h
Intel® Trace Hub	B6h



## 41.0 Testability

---

### JTAG:

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

### Intel® Trace Hub:

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

Intel® Trace Hub is a set of silicon features with supported software API. The primary purpose is to collect trace data from different sources in the system and combine them into a single output stream with time-correlated to each other. Intel® Trace Hub uses common hardware interface for collecting time-correlated system traces through standard destinations. Intel® Trace Hub adopts industry standard (MIPI\* STPv2) debug methodology for system debug and software development.

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

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

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

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

**Table 132. Acronyms**

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

**Table 133. References**

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

## 41.1 JTAG

This section provides information about Signal description and I/O Signal Planes and States.

### 41.1.1 Testability Signals

Signal Name	Description	Dir.	Buffer Type	Link Type	Availability
BPM#[3:0]	<b>Breakpoint and Performance Monitor Signals:</b> Outputs from the processor that indicate the status of breakpoints and programmable counters used for monitoring processor performance.	I/O	GTL	SE	N Processor Line
PROC_PRDY#	<b>Probe Mode Ready:</b> PROC_PRDY# is a processor output used by debug tools to determine processor debug readiness.	O	OD	SE	
PROC_PREQ#	<b>Probe Mode Request:</b> PROC_PREQ# is used by debug tools to request debug operation of the processor.	I	GTL	SE	
PROC_JTAG_TCK	<b>Test Clock:</b> This signal provides the clock input for the processor Test Bus (also known as the Test Access Port). This signal should be driven low or allowed to float during power on Reset.	I	GTL	SE	
PROC_JTAG_TDI	<b>Test Data In:</b> This signal transfers serial test data into the processor. This signal provides the serial input needed for JTAG specification support.	I	GTL	SE	
PROC_JTAG_TDO	<b>Test Data Out:</b> This signal transfers serial test data out of the processor. This signal provides the serial output needed for JTAG specification support.	O	OD	SE	

*continued...*

Signal Name	Description	Dir.	Buffer Type	Link Type	Availability
PROC_JTAG_TMS	<b>Test Mode Select:</b> A JTAG specification support signal used by debug tools.	I	GTL	SE	
PROC_JTAG_TRST#	<b>Test Reset:</b> Resets the Test Access Port (TAP) logic. This signal should be driven low during power on Reset.	I	GTL	SE	
DBG_PMODE					

### 41.1.2 Signal Description

Table 134. Testability Signals

Name	Type	Description
<b>PCH_JTAG_TCK</b>	I/O	<b>Test Clock Input (TCK):</b> The test clock input provides the clock for the JTAG test logic.
<b>PCH_JTAG_TMS</b>	I/OD	<b>Test Mode Select (TMS):</b> The signal is decoded by the Test Access Port (TAP) controller to control test operations.
<b>PCH_JTAG_TDI</b>	I/OD	<b>Test Data Input (TDI):</b> Serial test instructions and data are received by the test logic at TDI.
<b>PCH_JTAG_TDO</b>	I/OD	<b>Test Data Output (TDO):</b> TDO is the serial output for test instructions and data from the test logic defined in this standard.
<b>PCH_JTAGX</b>	I/O	This pin is used to support merged debug port topologies.
<b>DBG_PMODE</b>	O	ITP Power Mode Indicator. This signal is used to transmit processor and PCH power/reset information to the Debugger.

### 41.1.3 I/O Signal Planes and States

Table 135. Power Planes and States for Testability Signals

Signal Name	Power Plane	Resistors	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>PCH_JTAG_TCK</b>	Primary	Strong Internal Pull-Down	Driven Low	Driven Low	Driven Low	OFF
<b>PCH_JTAG_TMS</b>	Primary	Internal Pull-Up	Driven High	Driven High	Driven High	OFF
<b>PCH_JTAG_TDI</b>	Primary	Internal Pull-Up	Driven High	Driven High	Driven High	OFF
<b>PCH_JTAG_TDO</b>	Primary	External Pull-Up	Undriven	Undriven	Undriven	OFF
<b>PCH_JTAGX<sup>1</sup></b>	Primary	Internal Strong Pull-Up (as TDO Input), Internal Strong Pull-Down (as TCK Output)	Driven High	Driven High / Driven Low	Driven High / Driven Low	OFF
<b>DBG_PMODE</b>	Primary	Internal Pull-Up	Driven High	Driven High	Driven High	OFF

Notes: 1. This signal is used in common JTAG topology to take in last device's TDO to DCI. The only planned supported topology is the Shared Topology. Thus, this pin will operate as TCK mode.  
2. Reset reference for primary well pins is RSMRST#.

## 41.2 Boundry Scan Sideband Signals

This section provides information about Signal description for the BSSB signals.

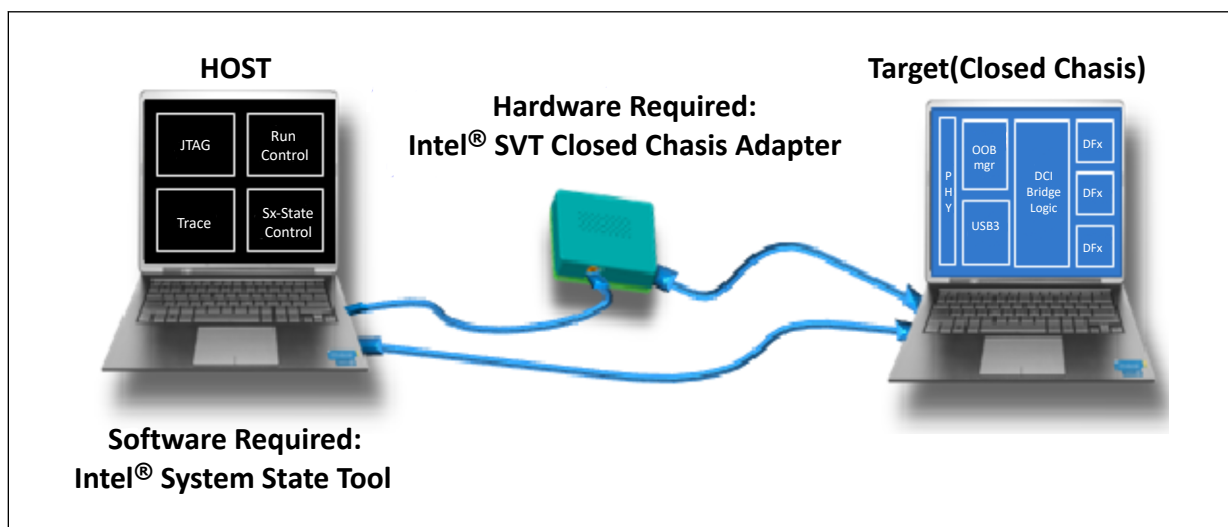
### 41.2.1 Signal Description

**Table 136. BSSB Signals**

Name	Type	Description
GPP_E19 / DDP1_CTRLCLK / <b>BSSB_LS0_TX</b>	I/O	Boundary Scan Sideband Low Speed Transmit 0 for debug purposes
GPP_E18 / DDP1_CTRLCLK / <b>BSSB_LS0_RX</b>	I/O	Boundary Scan Sideband Low Speed Receive 0 for debug purposes
GPP_E21 / DDP2_CTRLCLK / <b>BSSB_LS1_TX</b>	I/O	Boundary Scan Sideband Low Speed Transmit 1 for debug purposes
GPP_E20 / DDP2_CTRLCLK / <b>BSSB_LS1_RX</b>	I/O	Boundary Scan Sideband Low Speed Receive 1 for debug purposes
GPP_D10 / ISH_SPI_CLK / <b>BSSB_LS2_TX</b> / GSPI2_CLK	O	Boundary Scan Sideband Low Speed Transmit 2 for debug purposes
GPP_D9 / ISH_SPI_CS# / <b>BSSB_LS2_RX</b> / GSPI2_CS0#	I	Boundary Scan Sideband Low Speed Receive 2 for debug purposes
GPP_D12 / ISH_SPI_MOSI / <b>BSSB_LS3_TX</b> / GSPI2_MOSI	O	Boundary Scan Sideband Low Speed Transmit 3 for debug purposes
GPP_D11 / ISH_SPI_MISO / <b>BSSB_LS3_RX</b> / GSPI2_MISO	I	Boundary Scan Sideband Low Speed Receive 3 for debug purposes

### 41.3 Intel® Trace Hub (Intel® TH)

**Figure 39. Platform Setup with Intel® Trace Hub**



## 41.4 Direct Connect Interface (DCI)

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

---

### NOTE

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

---

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

- OOB Hosting DCI
- USB 3.2 Hosting DCI.DBC

Supported capabilities in DCI are:

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

Debug host software that support DCI are:

- Intel® System Studio (ISS)

### 41.4.1 Out Of Band (OOB) Hosting DCI

OOB was developed to provide an alternate path to convey controls and data to or from the EXI/DCI by connecting physically to the target through a USB 3.2 Gen 2x1 port. OOB provides an alternate side band path around the USB 3.2 controller, so that the embedded logic can be accessed, even when the USB 3.2 controller is not alive (such as in low power states) or is malfunctioning. This path does not rely on USB 3.2 Gen 2x1 protocol, link layer, or physical layer, because the xHCI functions are generally not available in such conditions. Instead, this path relies on a special adapter that was developed by Intel called the Intel® SVT Closed Chassis Adapter (CCA). It is a simple data transformation device. This adapter generates a OOB signaling protocol operating at up to 400 MHz and serializes data flowing through it. This adapter works together with debug host software and the embedded logic, contain a back-pressure scheme that makes both sides tolerant of overflow and starvation conditions, which is equivalent of USB 3.2 link layer. This path also uses native DCI packet protocol instead of USB 3.2 Gen 2x1 protocol. DCI.OOB - slower speed, CCA box needed. But survives S0ix and Sx states. Provides early boot access. Cannot tolerate re-driver circuits in its path.

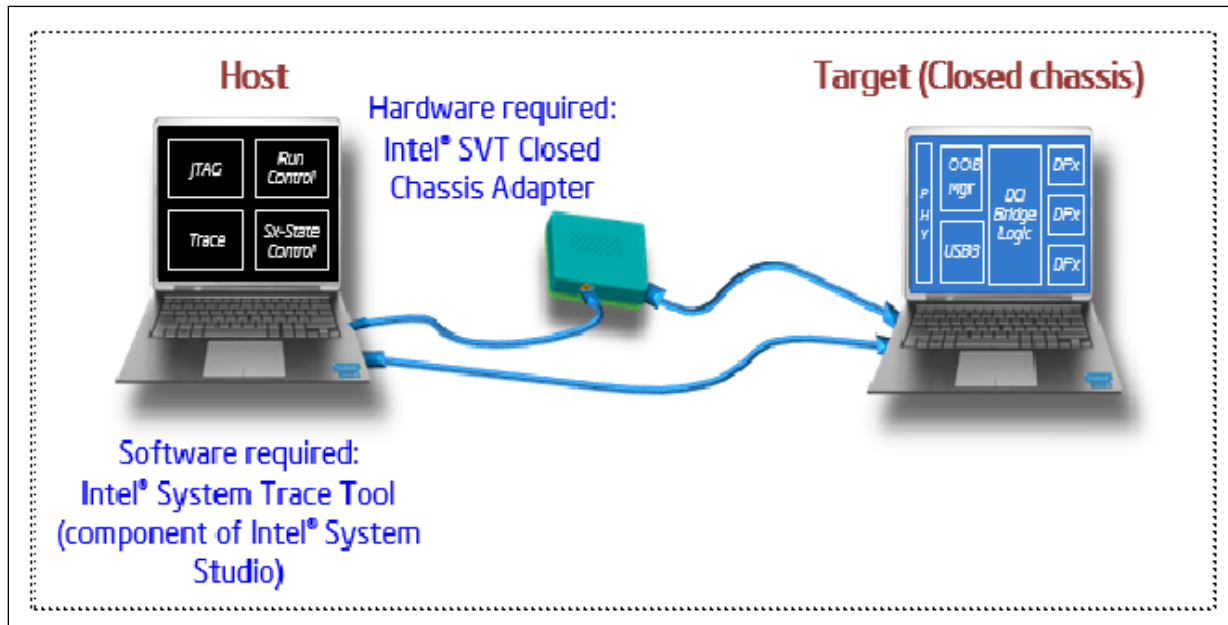
### 41.4.2 USB 3.2 Hosting DCI.DBC

It relies on Debug Class Devices (DbC) which is comprised of a set of logic that is bolted to the side of the xHCI host controller and enable the target to act the role of a USB device for debug purpose. This path uses the USB packet protocol layer, USB

layer flow control and USB physical layer at 5 GHz (for USB 3.2) and 480 MHz (for USB 2.0). DCI.DBC - Fast speed. USB 3.2 only works in S0. USB 2.0 survives S0ix and Sx states and provides early boot access.

### 41.4.3 Platform Setup

Figure 40. Platform Setup with DCI Connection



## 42.0 Digital Display Signals

**Table 137. Acronyms**

Acronyms	Description
eDP*	embedded Display Port*

### 42.1 Signal Description

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

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

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

The DDC (Digital Display Channel) bus is used for communication between the host system and display. Seven pairs of DDC (DDP\*\_CTRLCLK and DDP\*\_CTRLDATA) signals exist on the PCH that correspond to four digital ports on the processor. DDC follows I<sup>2</sup>C protocol.

The Hot-Plug Detect (HPD) signal serves as an interrupt request for the sink device for DisplayPort\* and HDMI\*. DDC and HPD signals are muxed with GPIO pads that can be independently configured to 1.8 V or 3.3 V via soft straps. Therefore, depending on soft strap settings for the corresponding GPIO pads, DDC and HPD signals can support either 1.8 V or 3.3 V.

**Table 138. Digital Display Signals**

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

Name	Type	Description
GPP_E22 / <b>DDPA_CTRLCLK</b> / DNX_FORCE_RELOAD	I/O	<b>Display Port A</b> : Control Clock.
GPP_E23 / <b>DDPA_CTRLDATA</b>	I/O	<b>Display Port A</b> : Control Data.
GPP_H15 / <b>DDPB_CTRLCLK</b> / PCIE_LINK_DOWN	I/O	<b>Display Port B</b> : Control Clock.
GPP_H17 / <b>DDPB_CTRLDATA</b>	I/O	<b>Display Port B</b> : Control Data.
GPP_E18 / <b>DDP1_CTRLCLK</b>	I/O	<b>Display Port 1</b> : Control Clock.
GPP_E19 / <b>DDP1_CTRLDATA</b>	I/O	<b>Display Port 1</b> : Control Data.
GPP_E20 / <b>DDP2_CTRLCLK</b>	I/O	<b>Display Port 2</b> : Control Clock.
GPP_E21 / <b>DDP2_CTRLDATA</b>	I/O	<b>Display Port 2</b> : Control Data.

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

**Table 139. Embedded DisplayPort\* (eDP\*) Backlight Control Signals**

Signal Name	Type	Description
<b>VDDEN</b>	0	<b>Primary eDP Panel power Enable</b> : Panel power control enable. This signal is used to control the VDC source of the panel logic.
<b>eDP_BKLTEN</b>	0	<b>Primary eDP Backlight Enable</b> : Panel backlight enable control for eDP. This signal is used to gate power into the backlight circuitry.
<b>eDP_BKLTCTL</b>	0	<b>Primary eDP Panel Backlight Brightness control</b> : Panel brightness control for eDP. This signal is used as the PWM Clock input signal.
GPP_A17 / <b>DISP_MISCC</b>	0	<b>Secondary eDP Panel power Enable</b> : Panel power control enable. This signal is used to control the VDC source of the panel logic.
GPP_E14 / DDSP_HPDA / <b>DISP_MISCA</b>	0	Display Miscellaneous Control Signal
GPP_A20 / DDSP_HPDA2 / <b>DISP_MISC2</b>	0	Display Miscellaneous Control Signal
GPP_A19 / DDSP_HPDA1 / <b>DISP_MISC1</b>	0	Display Miscellaneous Control Signal
GPP_A18 / DDSP_HPDB / <b>DISP_MISCB</b>	0	Display Miscellaneous Control Signal
GPP_A15 / USB_OC2# / DDSP_HPDA4 / <b>DISP_MISC4</b>	0	Display Miscellaneous Control Signal
GPP_A14 / USB_OC1# / DDSP_HPDA3 / <b>DISP_MISC3</b>	0	Display Miscellaneous Control Signal
<i>Note:</i> VDDEN, eDP_BKLTEN, eDP_BKLTCTL, DISP_MISCC can be left as no connect if eDP* is not used.		



## 42.3 Integrated Pull-Ups and Pull-Downs

**Table 140. Integrated Pull-Ups and Pull-Downs**

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

## 42.4 I/O Signal Planes and States

**Table 141. I/O Signal Planes and States**

Signal Name	Power Plane	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S3/S4/S5	Deep Sx
<b>DDSP_HPDA</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>DDSP_HPDB</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>DDSP_HPDA1</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>DDSP_HPDA2</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>DDSP_HPDA3</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>DDSP_HPDA4</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>DDPA_CTRLCLK</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>DDPA_CTRLDATA</b>	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
<b>DDPB_CTRLCLK</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>DDPB_CTRLDATA</b>	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
<b>DDP1_CTRLCLK</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>DDP1_CTRLDATA</b>	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
<b>DDP2_CTRLCLK</b>	Primary	Undriven	Undriven	Undriven	OFF
<b>DDP2_CTRLDATA</b>	Primary	Internal Pull-down	Driven Low	Internal Pull-down	OFF
<b>eDP_VDDEN</b>	Primary	Driven Low	Driven Low	Driven Low	OFF
<b>eDP_BKLTEN</b>	Primary	Driven Low	Driven Low	Driven Low	OFF
<b>eDP_BKLTCTL</b>	Primary	Driven Low	Driven Low	Driven Low	OFF
<b>DISP_MISCC</b>	Primary	Driven Low	Driven Low	Driven Low	OFF
<i>Note:</i> 1. Reset reference for primary well pins is RSMRST#.					

## 43.0 Miscellaneous Signals

### 43.1 Signal Description

Table 142. Signal Descriptions

Name	Type	Description
GPP_D0 / ISH_GP0 / <b>BK0</b> / SBK0	OD	<b>Blink BK 0:</b> This function provides the blink (or PWM) capability. The blink/PWM frequency and duty cycle is programmable through the PWM Control register. Refer to Volume 2 for details.
GPP_D1 / ISH_GP1 / <b>BK1</b> / SBK1	OD	<b>Blink BK 1:</b> This function provides the blink (or PWM) capability. The blink/PWM frequency and duty cycle is programmable through the PWM Control register. Refer to Volume 2 for details.
GPP_D2 / ISH_GP2 / <b>BK2</b> / SBK2	OD	<b>Blink BK 2:</b> This function provides the blink (or PWM) capability. The blink/PWM frequency and duty cycle is programmable through the PWM Control register. Refer to Volume 2 for details.
GPP_D3 / ISH_GP3/ <b>BK3</b> / SBK3	OD	<b>Blink BK 3:</b> This function provides the blink (or PWM) capability. The blink/PWM frequency and duty cycle is programmable through the PWM Control register. Refer to Volume 2 for details.
GPP_D4 / IMGCLKOUT0 / <b>BK4</b> / SBK4	OD	<b>Blink BK 4:</b> This function provides the blink (or PWM) capability. The blink/PWM frequency and duty cycle is programmable through the PWM Control register. Refer to Volume 2 for details.
GPP_E22 / DDPA_CTRLCLK / <b>DNX_FORCE_RELOAD</b>	I	<b>Download and Execute (DnX):</b> Intel® CSE ROM samples this pin any time ROM begins execution. This includes the following conditions: <ul style="list-style-type: none"> <li>• G3 Exit.</li> <li>• Sx, Mofx Exit.</li> <li>• Cold Reset(Host Reset with Power Cycle) Exit.</li> <li>• Warm Reset(Host Reset without Power Cycle) Exit if Intel® CSE was shutdown in Warm Reset.</li> </ul>
GPP_E0 / <b>SATAXPCIE0</b> / SATAGP0	I	<b>SATA port 0 or PCIe port mux select :</b> This is used to select SATA/PCIe function to support implementations like SATA Express.
GPP_A12 / <b>SATAXPCIE1</b> / SATAGP1	I	<b>SATA port 1 or PCIe port mux select :</b> This is used to select SATA/PCIe function to support implementations like SATA Express.
GPP_D0 / ISH_GP0 / BK0 / <b>SBK0</b>	OD	<b>Serial Blink SBK 0:</b> This function provides the capability to serialize POST or other messages on the pin to a serial monitor. The Serial Blink message is programmed through the Serial Blink Command/Status and Serial Blink Data registers. Refer to Volume 2 for details.
GPP_D1 / ISH_GP1 / BK1 / <b>SBK1</b>	OD	<b>Serial Blink SBK 1:</b> This function provides the capability to serialize POST or other messages on the pin to a serial monitor. The Serial Blink message is programmed through the Serial Blink Command/Status and Serial Blink Data registers. Refer to Volume 2 for details.
GPP_D2 / ISH_GP2 / BK2 / <b>SBK2</b>	OD	<b>Serial Blink SBK 2:</b> This function provides the capability to serialize POST or other messages on the pin to a serial monitor. The Serial Blink message is programmed through the Serial Blink Command/Status and Serial Blink Data registers. Refer to Volume 2 for details.

*continued...*

Name	Type	Description
GPP_D3 / ISH_GP3 / BK3 / <b>SBK3</b>	OD	<b>Serial Blink SBK 3:</b> This function provides the capability to serialize POST or other messages on the pin to a serial monitor. The Serial Blink message is programmed through the Serial Blink Command/Status and Serial Blink Data registers. Refer to Volume 2 for details.
GPP_D4 / IMGCLKOUT0 / BK4 / <b>SBK4</b>	OD	<b>Serial Blink SBK 4:</b> This function provides the capability to serialize POST or other messages on the pin to a serial monitor. The Serial Blink message is programmed through the Serial Blink Command/Status and Serial Blink Data registers. Refer to Volume 2 for details.
GPP_B15 / <b>TIME_SYNC0</b> / ISH_GP7	I	<b>Time Synchronization GPIO 0:</b> Timed GPIO event for time synchronization for interfaces that do not support time synchronization natively.
PROC_POPIRCOMP		50 Ohm±1% pulldown to ground
MPHY_RCOMP	In	100 ohm (+/- 1%) connected between MPHY_RCOMP and MPHY_RCOMP
MPHY_RCOMP	In	100 ohm (+/- 1%) connected between MPHY_RCOMP and MPHY_RCOMP
GPPC_RCOMP	InOut	Analog connection point for an external bias resistor to ground(200ohm((+/- 1%))
DMI_RCOMP		OPI Compensation (50 Ohm±1% pulldown to ground)
GPP_F5 / <b>MODEM_CLKREQ</b> / CRF_XTAL_CLKREQ	Out	CRF: Wake/activity request from SOC side.Optional PCM interfacewhen used with Discrete
GPP_H13 / I2C7_SCL / UART0_CTS# / <b>M2_SKT2_CFG3</b> / ISH_GP7B / SATA_DEVSLP1B	In	M.2 Socket 2 Configuration. Used to select SSD@PCIe/SATA, WWAN@PCIe/USB3/SSIC, etc. Refer to the table <b>Socket 2 Module Configuration</b> in the <a href="#">PCIe M.2 ECN</a> for details
GPP_H12 / I2C7_SDA / UART0_RTS# / <b>M2_SKT2_CFG2</b> / ISH_GP6B / SATA_DEVSLP0B	In	M.2 Socket 2 Configuration. Used to select SSD@PCIe/SATA, WWAN@PCIe/USB3/SSIC, etc. Refer to the table <b>Socket 2 Module Configuration</b> in the <a href="#">PCIe M.2 ECN</a> for details
GPP_H11 / UART0_TXD / <b>M2_SKT2_CFG1</b>	In	M.2 Socket 2 Configuration. Used to select SSD@PCIe/SATA, WWAN@PCIe/USB3/SSIC, etc. Refer to the table <b>Socket 2 Module Configuration</b> in the <a href="#">PCIe M.2 ECN</a> for details
GPP_H10 / UART0_RXD / <b>M2_SKT2_CFG0</b>	In	M.2 Socket 2 Configuration. Used to select SSD@PCIe/SATA, WWAN@PCIe/USB3/SSIC, etc. Refer to the table <b>Socket 2 Module Configuration</b> in the <a href="#">PCIe M.2 ECN</a> for details
GPP_B18 / <b>ADR_COMPLETE</b>	Out	Auto-DIMM Self Refresh complete indicator

## 43.2 Reset and Miscellaneous Signals

Signal Name	Description	Dir.	Buffer Type	Link Type
CFG[17:0]	<p>Configuration Signals: The CFG signals have a default value of '1' if not terminated on the board.</p> <p>Intel recommends placing test points on the board for CFG pins.</p> <ul style="list-style-type: none"> <li>CFG[1:0]: Reserved configuration lane.</li> <li>CFG[2]:</li> <li>CFG[3]: Reserved configuration lane.</li> <li>CFG[4]: Reserved</li> <li>CFG[5] Reserved configuration lanes.</li> </ul>	I	GTL	SE
<i>continued...</i>				

Signal Name	Description	Dir.	Buffer Type	Link Type
	<ul style="list-style-type: none"> <li>CFG[6]: Reserved configuration lanes.</li> <li>CFG[7]: Reserved configuration lanes.</li> <li>CFG[13:8]: Reserved configuration lanes.</li> <li>CFG[14]</li> <li>CFG[17:15]: Reserved configuration lanes.</li> </ul>			
CFG_RCOMP	Configuration Resistance Compensation	NA	NA	SE
EAR#	Stall reset sequence for early reset phases debug until deasserted: – 1 = (Default) Normal Operation; No stall. – 0 = Stall.	I	CMOS	SE
DRAM_RESET#	Memory Reset	O	CMOS	SE

### 43.3 Ground and Reserved Signals

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

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

Arbitrary connection of these signals to VCC, VDD2, VSS, or to any other signal (including each other) may result in component malfunction or incompatibility with future processors. Refer to the table below.

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

**Table 143. GND, RSVD, and NCTF Signals**

Signal Name	Description
Vss	<b>Ground:</b> Processor ground node
Vss_NCTF	<b>Non-Critical To Function:</b> These signals are for package mechanical reliability and should not be connected on the board.
RSVD	<b>Reserved:</b> All signals that are RSVD should not be connected on the board.
RSVD_NCTF	<b>Reserved Non-Critical To Function:</b> RSVD_NCTF should not be connected on the board.
RSVD_TP	<b>Test Point:</b> Intel recommends to route each RSVD_TP to an accessible test point. Intel may require these test points for platform specific debug. Leaving these test points inaccessible could delay debug by Intel.

## 43.4 Integrated Pull-Ups and Pull-Downs

**Table 144. Integrated Pull-Ups and Pull-Downs**

Signal	Resistor Type	Value
SATAXPCIE0	Pull-down	20 kohm
SATAXPCIE1	Pull-down	20 kohm

## 43.5 I/O Signal Planes and States

None

## 43.6 Processor Internal Pull-Up / Pull-Down Terminations

Signal Name	Pull Up/Pull Down	Rail	Value ( $\Omega$ )
BPM_N[3:0]	Pull Up/Pull Down	VCCANA	1
PROC_PREQ#	Pull Up	VCC1p05_PROC	1
PROC_TDI	Pull Up	VCC1p05_PROC	1
PROC_TMS	Pull Up	VCC1p05_PROC	1
PROC_TRST#	Pull Down	VCC1p05_PROC	1
PROC_TCK	Pull Down	VCC1p05_PROC	1
CFG[17:0]	Pull Up	VCCANA	1

## **44.0 On Package Interface (OPI)**

---

### **44.1 On Package Interface (OPI)**

#### **44.1.1 OPI Support**

The processor communicates with the PCH using an internal interconnect BUS named OPI.

#### **44.1.2 Functional Description**

OPI operates at 4 GT/s bus rate.

## 45.0 embedded Multi Media Card (eMMC\*)

---

The eMMC\* is a universal data storage and communication media. It is designed to cover a wide area of applications such as smart phones, tablets, computers, cameras, and so on. PCH supports only 1.8 V operating devices and PCH supports eMMC\* version 5.1.

### 45.1 Features Supported

- HW Command Queuing support compliant to eMMC\* v5.1 specification
- Support enhanced Strobe for HS400 mode @1.8 V
- Both ADMA2/DMA and Non-DMA mode of operation
- Transfers the data in 1 bit, 4 bit and 8 bit mode
- Support 64b address
- Cyclic Redundancy Check CRC7 for command and CRC16 for data integrity
- Support for Tx Path tuning and retention of DLL delay values

### 45.2 Functional Description

The Controller handles eMMC\* Protocol at transmission, packing data, adding cyclic redundancy check (CRC), start/end bit, and checking for transaction format correctness. Main supported features are listed below.

The eMMC\* main use case is to connect an on board external storage device.

#### eMMC\* 5.1 Command Queuing

Command Queuing (CQ) definition for eMMC\* includes new commands for issuing tasks to the device, for ordering the execution of previously issued tasks and for additional task management function. The host controller with CQ can queue up to 32 commands to the device and the device selects and indicates one of the queued commands to host for service.

The host controller implements additional logic for handling a door-bell based DMA for the 32 descriptor / task list and manages the entire CQ flow which includes:

- Fetch and send the tasks/commands to device using existing logic
- Maintains context of each queued command
- Periodically read the device queue status and indicates completion of task to SW.
- Implements interrupt coalescing to reduce burden on software ISR.

#### eMMC\* 5.1 Enhanced Strobe

Enhanced Strobe Mode for HS400 improves upon the HS400 mode interface speed increase that was first defined in eMMC\* version 5.0, by facilitating faster synchronization between the host and the device.

Refer JEDEC eMMC\* 5.1 specification for additional information.

### eMMC\* Working Modes

eMMC* Mode	Data Rate	Clock Frequency	Max. Data Throughput
Compatibility	Single	0 – 25 MHz	25 MB/s
High Speed SDR	Single	0 – 25 MHz	25 MB/s
High Speed DDR	Dual	0 – 25 MHz	50 MB/s
HS200	Single	0 - 200 MHz	200 MB/s
HS400	Dual	0 - 200 MHz	400 MB/s

## 45.3 Signal Description

Name	Type	Description
EMMC_CMD	I/O	eMMC* Command/Response
EMMC_DATA[7:0]	I/O	eMMC* Data [7:0]
EMMC_RCLK	I	eMMC* Receive Clock
EMMC_CLK	O	eMMC* Clock
EMMC_RCOMP	I/O	eMMC* Compensation (200 Ohm +/- 1 % pull down to ground)
EMMC_RESET#	O	eMMC* Device/Media Reset

## 45.4 I/O Signal Planes and States

Signal Name	Power Well	During Reset <sup>1</sup>	Immediately after Reset <sup>1</sup>	S0/S3/S4/S5	Deep Sx
EMMC_DATA[7:0]	Primary	Undriven	Undriven	Undriven	OFF
EMMC_RCLK	Primary	Undriven	Undriven	Undriven	OFF
EMMC_CLK	Primary	Undriven	Undriven	Undriven	OFF
EMMC_CMD	Primary	Undriven	Undriven	Undriven	OFF
EMMC_RCOMP	Primary	Undriven	Undriven	Undriven	OFF
EMMC_RESET#	Primary	Undriven	Undriven	Undriven	OFF

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