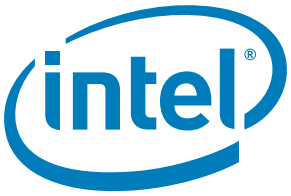


# Intel® NM10 Family Express Chipset

## Datasheet

***December 2009***

Document Number: 322896-001

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

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

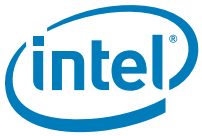
Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

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

Intel, Intel SpeedStep, and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Copyright © 2009, Intel Corporation



# Contents


**1 Introduction** ..... 30

1.1 Intel NM10 Family Express Chipset Feature Support ..... 31
1.2 Content Layout ..... 34
1.3 Functions and capabilities ..... 36

**2 Signal Description** ..... 43

2.1 Direct Media Interface (DMI) to Host Controller ..... 45
2.2 PCI Express* ..... 45
2.3 Platform LAN Connect Interface ..... 45
2.4 EEPROM Interface ..... 46
2.5 Firmware Hub Interface ..... 46
2.6 PCI Interface ..... 47
2.7 Serial ATA Interface ..... 49
2.8 LPC Interface ..... 50
2.9 Interrupt Interface ..... 50
2.10 USB Interface ..... 51
2.11 Power Management Interface ..... 52
2.12 Processor Interface ..... 54
2.13 SMBus Interface ..... 56
2.14 System Management Interface ..... 56
2.15 Real Time Clock Interface ..... 56
2.16 Other Clocks ..... 57
2.17 Miscellaneous Signals ..... 57
2.18 Intel HD Audio Link ..... 58
2.19 Serial Peripheral Interface (SPI) ..... 59
2.20 General Purpose I/O Signals ..... 59
2.21 Power and Ground ..... 60
2.22 Pin Straps ..... 61
2.22.1 Functional Straps ..... 61
2.22.2 External RTC Circuitry ..... 63
2.23 Device and Revision ID Table ..... 64

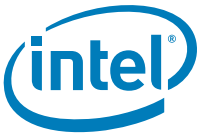
**3 Pin States** ..... 65

3.1 Integrated Pull-Ups and Pull-Downs ..... 65
3.2 Output and I/O Signals Planes and States ..... 66
3.3 Power Planes for Input Signals ..... 71

**4 Chipset and System Clock Domains** ..... 74

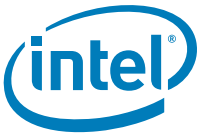
**5 Functional Description** ..... 76

5.1 PCI-to-PCI Bridge (D30:F0) ..... 76
5.1.1 PCI Bus Interface ..... 76
5.1.2 PCI Bridge As an Initiator ..... 76
5.1.3 Parity Error Detection and Generation ..... 78
5.1.4 PCIRST# ..... 79
5.1.5 Peer Cycles ..... 79
5.1.6 PCI-to-PCI Bridge Model ..... 80
5.1.7 IDSEL to Device Number Mapping ..... 80
5.1.8 Standard PCI Bus Configuration Mechanism ..... 80




5.2 PCI Express* Root Ports (D28:F0,F1,F2,F3) ..... 81
5.2.1 Interrupt Generation ..... 81
5.2.2 Power Management ..... 82
5.2.3 SERR# Generation ..... 83
5.2.4 Hot-Plug ..... 84
5.3 LAN Controller (B1:D8:F0) ..... 86
5.3.1 LAN Controller PCI Bus Interface ..... 86
5.3.2 Serial EEPROM Interface ..... 91
5.3.3 CSMA/CD Unit ..... 91
5.3.4 Media Management Interface ..... 92
5.3.5 TCO Functionality ..... 92
5.4 Alert Standard Format (ASF) ..... 94
5.4.1 ASF Management Solution Features/Capabilities ..... 95
5.4.2 ASF Hardware Support ..... 96
5.4.3 ASF Software Support ..... 97
5.5 LPC Bridge (w/ System and Management Functions) (D31:F0) ..... 98
5.5.1 LPC Interface ..... 98
5.5.2 SERR# Generation ..... 103
5.6 DMA Operation (D31:F0) ..... 104
5.6.1 Channel Priority ..... 105
5.6.2 Address Compatibility Mode ..... 105
5.6.3 Summary of DMA Transfer Sizes ..... 106
5.6.4 Autoinitialize ..... 106
5.6.5 Software Commands ..... 107
5.7 LPC DMA ..... 107
5.7.1 Asserting DMA Requests ..... 107
5.7.2 Abandoning DMA Requests ..... 108
5.7.3 General Flow of DMA Transfers ..... 108
5.7.4 Terminal Count ..... 109
5.7.5 Verify Mode ..... 109
5.7.6 DMA Request Deassertion ..... 109
5.7.7 SYNC Field / LDRQ# Rules ..... 110
5.8 8254 Timers (D31:F0) ..... 111
5.8.1 Timer Programming ..... 112
5.8.2 Reading from the Interval Timer ..... 113
5.9 8259 Interrupt Controllers (PIC) (D31:F0) ..... 114
5.9.1 Interrupt Handling ..... 116
5.9.2 Initialization Command Words (ICWx) ..... 117
5.9.3 Operation Command Words (OCW) ..... 118
5.9.4 Modes of Operation ..... 118
5.9.5 Masking Interrupts ..... 121
5.9.6 Steering PCI Interrupts ..... 121
5.10 Advanced Programmable Interrupt Controller (APIC) (D31:F0) ..... 122
5.10.1 Interrupt Handling ..... 122
5.10.2 Interrupt Mapping ..... 122
5.10.3 PCI / PCI Express* Message-Based Interrupts ..... 123
5.10.4 Front Side Bus Interrupt Delivery ..... 124
5.11 Serial Interrupt (D31:F0) ..... 126
5.11.1 Start Frame ..... 126
5.11.2 Data Frames ..... 126
5.11.3 Stop Frame ..... 127
5.11.4 Specific Interrupts Not Supported via SERIRQ ..... 127
5.11.5 Data Frame Format ..... 127

5.12 Real Time Clock (D31:F0) ..... 129
5.12.1 Update Cycles ..... 129
5.12.2 Interrupts ..... 130
5.12.3 Lockable RAM Ranges ..... 130
5.12.4 Century Rollover ..... 130
5.12.5 Clearing Battery-Backed RTC RAM ..... 131
5.13 Processor Interface (D31:F0) ..... 132
5.13.1 Processor Interface Signals ..... 133
5.13.2 Dual-Processor Issues (Nettop Only) ..... 135
5.14 Power Management (D31:F0) ..... 136
5.14.1 Features ..... 136
5.14.2 Chipset and System Power States ..... 137
5.14.3 System Power Planes ..... 139
5.14.4 SMI#/SCI Generation ..... 140
5.14.5 Dynamic Processor Clock Control ..... 142
5.14.6 Dynamic PCI Clock Control (Netbook Only) ..... 145
5.14.7 Sleep States ..... 147
5.14.8 Thermal Management ..... 150
5.14.9 Event Input Signals and Their Usage ..... 152
5.14.10 ALT Access Mode ..... 155
5.14.11 System Power Supplies, Planes, and Signals ..... 158
5.14.12 Clock Generators ..... 161
5.14.13 Legacy Power Management Theory of Operation ..... 162
5.15 System Management (D31:F0) ..... 163
5.15.1 Theory of Operation ..... 163
5.15.2 Heartbeat and Event Reporting via SMBus ..... 164
5.16 SATA Host Controller (D31:F2) ..... 168
5.16.1 Theory of Operation ..... 170
5.16.2 SATA Swap Bay Support ..... 171
5.16.3 Power Management Operation ..... 171
5.16.4 SATA LED ..... 173
5.16.5 AHCI Operation ..... 173
5.17 High Precision Event Timers ..... 174
5.17.1 Timer Accuracy ..... 174
5.17.2 Interrupt Mapping ..... 175
5.17.3 Periodic vs. Non-Periodic Modes ..... 175
5.17.4 Enabling the Timers ..... 176
5.17.5 Interrupt Levels ..... 176
5.17.6 Handling Interrupts ..... 176
5.17.7 Issues Related to 64-Bit Timers with 32-Bit Processors ..... 177
5.18 USB UHCI Host Controllers (D29:F0, F1, F2, and F3) ..... 177
5.18.1 Data Structures in Main Memory ..... 177
5.18.2 Data Transfers to/from Main Memory ..... 178
5.18.3 Data Encoding and Bit Stuffing ..... 178
5.18.4 Bus Protocol ..... 178
5.18.5 Packet Formats ..... 179
5.18.6 USB Interrupts ..... 179
5.18.7 USB Power Management ..... 182
5.18.8 USB Legacy Keyboard Operation ..... 182
5.19 USB EHCI Host Controller (D29:F7) ..... 185
5.19.1 EHC Initialization ..... 185
5.19.2 Data Structures in Main Memory ..... 186
5.19.3 USB 2.0 Enhanced Host Controller DMA ..... 186




5.19.4 Data Encoding and Bit Stuffing ..... 187
5.19.5 Packet Formats ..... 187
5.19.6 USB 2.0 Interrupts and Error Conditions ..... 187
5.19.7 USB 2.0 Power Management ..... 188
5.19.8 Interaction with UHCI Host Controllers ..... 190
5.19.9 USB 2.0 Legacy Keyboard Operation ..... 193
5.19.10 USB 2.0 Based Debug Port ..... 193
5.20 SMBus Controller (D31:F3) ..... 199
5.20.1 Host Controller ..... 199
5.20.2 Bus Arbitration ..... 204
5.20.3 Bus Timing ..... 205
5.20.4 Interrupts / SMI# ..... 205
5.20.5 SMBALERT# ..... 207
5.20.6 SMBus CRC Generation and Checking ..... 207
5.20.7 SMBus Slave Interface ..... 207
5.21 Intel HD Audio Overview ..... 213
5.22 Serial Peripheral Interface (SPI) ..... 214
5.22.1 Flash Device Configurations ..... 214
5.22.2 SPI Device Compatibility Requirements ..... 214
5.22.3 Chipset Compatible Command Set ..... 215
5.22.4 Flash Protection ..... 216
5.23 Feature Capability Mechanism ..... 217

**6 Ballout Definition ..... 218**

6.1 Chipset Ballout, Signal, and Mechanical Document ..... 218
6.2 Chipset Ballout ..... 218

**7 Chipset Package Information ..... 225**

**8 Electrical Characteristics ..... 227**

8.1 Thermal Specifications ..... 227
8.2 Absolute Maximum Ratings ..... 227
8.3 DC Characteristics ..... 228
8.4 AC Characteristics ..... 235
8.5 Timing Diagrams ..... 244

**9 Register and Memory Mapping ..... 258**

9.1 PCI Devices and Functions ..... 258
9.2 PCI Configuration Map ..... 259
9.3 I/O Map ..... 260
9.3.1 Fixed I/O Address Ranges ..... 260
9.3.2 Variable I/O Decode Ranges ..... 262
9.4 Memory Map ..... 264
9.4.1 Boot-Block Update Scheme ..... 265

**10 Chipset Configuration Registers ..... 267**

10.1 Chipset Configuration Registers (Memory Space) ..... 267
10.1.1 VCH—Virtual Channel Capability Header Register ..... 269
10.1.2 VCAP1—Virtual Channel Capability #1 Register ..... 269
10.1.3 VCAP2—Virtual Channel Capability #2 Register ..... 270
10.1.4 PVC—Port Virtual Channel Control Register ..... 270
10.1.5 PVS—Port Virtual Channel Status Register ..... 270
10.1.6 V0CAP—Virtual Channel 0 Resource Capability Register ..... 270
10.1.7 V0CTL—Virtual Channel 0 Resource Control Register ..... 271

10.1.8 V0STS—Virtual Channel 0 Resource Status Register ..... 271
10.1.9 V1CAP—Virtual Channel 1 Resource Capability Register ..... 272
10.1.10 V1CTL—Virtual Channel 1 Resource Control Register ..... 272
10.1.11 V1STS—Virtual Channel 1 Resource Status Register ..... 273
10.1.12 RCTCL—Root Complex Topology Capabilities List Register ..... 273
10.1.13 ESD—Element Self Description Register ..... 273
10.1.14 ULD—Upstream Link Descriptor Register ..... 274
10.1.15 ULBA—Upstream Link Base Address Register ..... 274
10.1.16 RP1D—Root Port 1 Descriptor Register ..... 274
10.1.17 RP1BA—Root Port 1 Base Address Register ..... 275
10.1.18 RP2D—Root Port 2 Descriptor Register ..... 275
10.1.19 RP2BA—Root Port 2 Base Address Register ..... 275
10.1.20 RP3D—Root Port 3 Descriptor Register ..... 276
10.1.21 RP3BA—Root Port 3 Base Address Register ..... 276
10.1.22 RP4D—Root Port 4 Descriptor Register ..... 276
10.1.23 RP4BA—Root Port 4 Base Address Register ..... 277
10.1.24 HDD—Intel HD Audio Descriptor Register ..... 277
10.1.25 HDBA—Intel HD Audio Base Address Register ..... 277
10.1.26 ILCL—Internal Link Capabilities List Register ..... 278
10.1.27 LCAP—Link Capabilities Register ..... 278
10.1.28 LCTL—Link Control Register ..... 279
10.1.29 LSTS—Link Status Register ..... 279
10.1.30 RPC—Root Port Configuration Register ..... 280
10.1.31 RPFN—Root Port Function Number for PCI Express Root Ports ..... 281
10.1.32 TRSR—Trap Status Register ..... 282
10.1.33 TRCR—Trapped Cycle Register ..... 282
10.1.34 TWDR—Trapped Write Data Register ..... 282
10.1.35 IOTRn — I/O Trap Register (0-3) ..... 283
10.1.36 TCTL—TCO Configuration Register ..... 284
10.1.37 D31IP—Device 31 Interrupt Pin Register ..... 284
10.1.38 D30IP—Device 30 Interrupt Pin Register ..... 285
10.1.39 D29IP—Device 29 Interrupt Pin Register ..... 285
10.1.40 D28IP—Device 28 Interrupt Pin Register ..... 286
10.1.41 D27IP—Device 27 Interrupt Pin Register ..... 287
10.1.42 D31IR—Device 31 Interrupt Route Register ..... 288
10.1.43 D30IR—Device 30 Interrupt Route Register ..... 289
10.1.44 D29IR—Device 29 Interrupt Route Register ..... 290
10.1.45 D28IR—Device 28 Interrupt Route Register ..... 292
10.1.46 D27IR—Device 27 Interrupt Route Register ..... 293
10.1.47 OIC—Other Interrupt Control Register ..... 294
10.1.48 RC—RTC Configuration Register ..... 295
10.1.49 HPTC—High Precision Timer Configuration Register ..... 295
10.1.50 GCS—General Control and Status Register ..... 296
10.1.51 BUC—Backed Up Control Register ..... 298
10.1.52 FD—Function Disable Register ..... 298
10.1.53 CG—Clock Gating (Netbook only) ..... 300

**11 LAN Controller Registers (B1:D8:F0) ..... 302**

11.1 PCI Configuration Registers (LAN Controller—B1:D8:F0) ..... 302
11.1.1 VID—Vendor Identification Register (LAN Controller—B1:D8:F0) ..... 303

11.1.2 DID—Device Identification Register (LAN Controller—B1:D8:F0) ..... 303
11.1.3 PCICMD—PCI Command Register (LAN Controller—B1:D8:F0) ..... 304
11.1.4 PCISTS—PCI Status Register (LAN Controller—B1:D8:F0) ..... 305
11.1.5 RID—Revision Identification Register (LAN Controller—B1:D8:F0) ..... 306
11.1.6 SCC—Sub Class Code Register (LAN Controller—B1:D8:F0) ..... 306
11.1.7 BCC—Base-Class Code Register (LAN Controller—B1:D8:F0) ..... 306
11.1.8 CLS—Cache Line Size Register (LAN Controller—B1:D8:F0) ..... 307
11.1.9 PMLT—Primary Master Latency Timer Register (LAN Controller—B1:D8:F0) ..... 307
11.1.10 HEADTYP—Header Type Register (LAN Controller—B1:D8:F0) ..... 307
11.1.11 CSR_MEM_BASE — CSR Memory-Mapped Base Address Register (LAN Controller—B1:D8:F0) ..... 308
11.1.12 CSR_IO_BASE — CSR I/O-Mapped Base Address Register (LAN Controller—B1:D8:F0) ..... 308
11.1.13 SVID — Subsystem Vendor Identification (LAN Controller—B1:D8:F0) ..... 309
11.1.14 SID — Subsystem Identification (LAN Controller—B1:D8:F0) ..... 309
11.1.15 CAP_PTR — Capabilities Pointer (LAN Controller—B1:D8:F0) ..... 309
11.1.16 INT_LN — Interrupt Line Register (LAN Controller—B1:D8:F0) ..... 310
11.1.17 INT_PN — Interrupt Pin Register (LAN Controller—B1:D8:F0) ..... 310
11.1.18 MIN_GNT — Minimum Grant Register (LAN Controller—B1:D8:F0) ..... 310
11.1.19 MAX_LAT — Maximum Latency Register (LAN Controller—B1:D8:F0) ..... 310
11.1.20 CAP_ID — Capability Identification Register (LAN Controller—B1:D8:F0) ..... 311
11.1.21 NXT_PTR — Next Item Pointer (LAN Controller—B1:D8:F0) ..... 311
11.1.22 PM_CAP — Power Management Capabilities (LAN Controller—B1:D8:F0) ..... 311
11.1.23 PMCSR — Power Management Control/ Status Register (LAN Controller—B1:D8:F0) ..... 312
11.1.24 PCIDATA — PCI Power Management Data Register (LAN Controller—B1:D8:F0) ..... 313
11.2 LAN Control / Status Registers (CSR) (LAN Controller—B1:D8:F0) ..... 313
11.2.1 SCB_STA—System Control Block Status Word Register (LAN Controller—B1:D8:F0) ..... 314
11.2.2 SCB_CMD—System Control Block Command Word Register (LAN Controller—B1:D8:F0) ..... 316
11.2.3 SCB_GENPNT—System Control Block General Pointer Register (LAN Controller—B1:D8:F0) ..... 318
11.2.4 PORT—PORT Interface Register (LAN Controller—B1:D8:F0) ..... 318

11.2.5 EEPROM_CNTL—EEPROM Control Register (LAN Controller—B1:D8:F0) ..... 320
11.2.6 MDI_CNTL—Management Data Interface (MDI) Control Register (LAN Controller—B1:D8:F0) ..... 320
11.2.7 REC_DMA_BC—Receive DMA Byte Count Register (LAN Controller—B1:D8:F0) ..... 321
11.2.8 EREC_INTR—Early Receive Interrupt Register (LAN Controller—B1:D8:F0) ..... 321
11.2.9 FLOW_CNTL—Flow Control Register (LAN Controller—B1:D8:F0) ..... 322
11.2.10 PMDR—Power Management Driver Register (LAN Controller—B1:D8:F0) ..... 323
11.2.11 GENCNTL—General Control Register (LAN Controller—B1:D8:F0) ..... 324
11.2.12 GENSTA—General Status Register (LAN Controller—B1:D8:F0) ..... 324
11.2.13 SMB_PCI—SMB via PCI Register (LAN Controller—B1:D8:F0) ..... 325
11.2.14 Statistical Counters (LAN Controller—B1:D8:F0) ..... 325
11.3 ASF Configuration Registers (LAN Controller—B1:D8:F0) ..... 328
11.3.1 ASF_RID—ASF Revision Identification Register (LAN Controller—B1:D8:F0) ..... 329
11.3.2 SMB_CNTL—SMBus Control Register (LAN Controller—B1:D8:F0) ..... 329
11.3.3 ASF_CNTL—ASF Control Register (LAN Controller—B1:D8:F0) ..... 330
11.3.4 ASF_CNTL_EN—ASF Control Enable Register (ASF Controller—B1:D8:F0) ..... 331
11.3.5 ENABLE—Enable Register (ASF Controller—B1:D8:F0) ..... 331
11.3.6 APM—APM Register (ASF Controller—B1:D8:F0) ..... 332
11.3.7 WTIM_CONF—Watchdog Timer Configuration Register (ASF Controller—B1:D8:F0) ..... 332
11.3.8 HEART_TIM—Heartbeat Timer Register (ASF Controller—B1:D8:F0) ..... 333
11.3.9 RETRAN_INT—Retransmission Interval Register (ASF Controller—B1:D8:F0) ..... 333
11.3.10 RETRAN_PCL—Retransmission Packet Count Limit ..... 334
11.3.11 ASF_WTIM1—ASF Watchdog Timer 1 Register (ASF Controller—B1:D8:F0) ..... 334
11.3.12 ASF_WTIM2—ASF Watchdog Timer 2 Register (ASF Controller—B1:D8:F0) ..... 334
11.3.13 PET_SEQ1—PET Sequence 1 Register (ASF Controller—B1:D8:F0) ..... 334
11.3.14 PET_SEQ2—PET Sequence 2 Register (ASF Controller—B1:D8:F0) ..... 335
11.3.15 STA—Status Register (ASF Controller—B1:D8:F0) ..... 335
11.3.16 FOR_ACT—Forced Actions Register (ASF Controller—B1:D8:F0) ..... 336
11.3.17 RMCP_SNUM—RMCP Sequence Number Register (ASF Controller—B1:D8:F0) ..... 337

11.3.18 SP_MODE—Special Modes Register (ASF Controller—B1:D8:F0) ..... 337
11.3.19 INPOLL_TCONF—Inter-Poll Timer Configuration Register (ASF Controller—B1:D8:F0) ..... 337
11.3.20 PHIST_CLR—Poll History Clear Register (ASF Controller—B1:D8:F0) ..... 338
11.3.21 PMSK1—Polling Mask 1 Register (ASF Controller—B1:D8:F0) ..... 339
11.3.22 PMSK2—Polling Mask 2 Register (ASF Controller—B1:D8:F0) ..... 339
11.3.23 PMSK3—Polling Mask 3 Register (ASF Controller—B1:D8:F0) ..... 339
11.3.24 PMSK4—Polling Mask 4 Register (ASF Controller—B1:D8:F0) ..... 339
11.3.25 PMSK5—Polling Mask 5 Register (ASF Controller—B1:D8:F0) ..... 340
11.3.26 PMSK6—Polling Mask 6 Register (ASF Controller—B1:D8:F0) ..... 340
11.3.27 PMSK7—Polling Mask 7 Register (ASF Controller—B1:D8:F0) ..... 340
11.3.28 PMSK8—Polling Mask 8 Register (ASF Controller—B1:D8:F0) ..... 341

**12 PCI-to-PCI Bridge Registers (D30:F0) ..... 342**

12.1 PCI Configuration Registers (D30:F0) ..... 342
12.1.1 VID— Vendor Identification Register (PCI-PCI—D30:F0) ..... 343
12.1.2 DID— Device Identification Register (PCI-PCI—D30:F0) ..... 343
12.1.3 PCICMD—PCI Command (PCI-PCI—D30:F0) ..... 343
12.1.4 PSTS—PCI Status Register (PCI-PCI—D30:F0) ..... 344
12.1.5 RID—Revision Identification Register (PCI-PCI—D30:F0) ..... 346
12.1.6 CC—Class Code Register (PCI-PCI—D30:F0) ..... 346
12.1.7 PMLT—Primary Master Latency Timer Register (PCI-PCI—D30:F0) ..... 347
12.1.8 HEADTYP—Header Type Register (PCI-PCI—D30:F0) ..... 347
12.1.9 BNUM—Bus Number Register (PCI-PCI—D30:F0) ..... 347
12.1.10 SMLT—Secondary Master Latency Timer Register (PCI-PCI—D30:F0) ..... 348
12.1.11 IOBASE_LIMIT—I/O Base and Limit Register (PCI-PCI—D30:F0) ..... 348
12.1.12 SECSTS—Secondary Status Register (PCI-PCI—D30:F0) ..... 348
12.1.13 MEMBASE_LIMIT—Memory Base and Limit Register (PCI-PCI—D30:F0) ..... 349
12.1.14 PREF_MEM_BASE_LIMIT—Prefetchable Memory Base and Limit Register (PCI-PCI—D30:F0) ..... 350
12.1.15 PMBU32—Prefetchable Memory Base Upper 32 Bits Register (PCI-PCI—D30:F0) ..... 350
12.1.16 PMLU32—Prefetchable Memory Limit Upper 32 Bits Register (PCI-PCI—D30:F0) ..... 350
12.1.17 CAPP—Capability List Pointer Register (PCI-PCI—D30:F0) ..... 351
12.1.18 INTR—Interrupt Information Register (PCI-PCI—D30:F0) ..... 351
12.1.19 BCTRL—Bridge Control Register (PCI-PCI—D30:F0) ..... 351
12.1.20 SPDH—Secondary PCI Device Hiding Register (PCI-PCI—D30:F0) ..... 353
12.1.21 DTC—Delayed Transaction Control Register (PCI-PCI—D30:F0) ..... 354

12.1.22 BPS—Bridge Proprietary Status Register (PCI-PCI—D30:F0) ..... 355
12.1.23 BPC—Bridge Policy Configuration Register (PCI-PCI—D30:F0) ..... 356
12.1.24 SVCAP—Subsystem Vendor Capability Register (PCI-PCI—D30:F0) ..... 357
12.1.25 SVID—Subsystem Vendor IDs Register (PCI-PCI—D30:F0) ..... 357

**13 LPC Interface Bridge Registers (D31:F0)** ..... 358

13.1 PCI Configuration Registers (LPC I/F—D31:F0) ..... 358
13.1.1 VID—Vendor Identification Register (LPC I/F—D31:F0) ..... 359
13.1.2 DID—Device Identification Register (LPC I/F—D31:F0) ..... 359
13.1.3 PCICMD—PCI COMMAND Register (LPC I/F—D31:F0) ..... 360
13.1.4 PCISTS—PCI Status Register (LPC I/F—D31:F0) ..... 360
13.1.5 RID—Revision Identification Register (LPC I/F—D31:F0) ..... 361
13.1.6 PI—Programming Interface Register (LPC I/F—D31:F0) ..... 361
13.1.7 SCC—Sub Class Code Register (LPC I/F—D31:F0) ..... 361
13.1.8 BCC—Base Class Code Register (LPC I/F—D31:F0) ..... 362
13.1.9 PLT—Primary Latency Timer Register (LPC I/F—D31:F0) ..... 362
13.1.10 HEADTYP—Header Type Register (LPC I/F—D31:F0) ..... 362
13.1.11 SS—Sub System Identifiers Register (LPC I/F—D31:F0) ..... 362
13.1.12 CAPP—Capability List Pointer (LPC I/F—D31:F0) ..... 363
13.1.13 PMBASE—ACPI Base Address Register (LPC I/F—D31:F0) ..... 363
13.1.14 ACPI_CNTL—ACPI Control Register (LPC I/F — D31:F0) ..... 363
13.1.15 GPIOBASE—GPIO Base Address Register (LPC I/F — D31:F0) ..... 364
13.1.16 GC—GPIO Control Register (LPC I/F — D31:F0) ..... 364
13.1.17 PIRQ[n]_ROUT—PIRQ[A,B,C,D] Routing Control Register (LPC I/F—D31:F0) ..... 365
13.1.18 SIRQ_CNTL—Serial IRQ Control Register (LPC I/F—D31:F0) ..... 365
13.1.19 PIRQ[n]_ROUT—PIRQ[E,F,G,H] Routing Control Register (LPC I/F—D31:F0) ..... 366
13.1.20 LPC_I/O_DEC—I/O Decode Ranges Register (LPC I/F—D31:F0) ..... 367
13.1.21 LPC_EN—LPC I/F Enables Register (LPC I/F—D31:F0) ..... 368
13.1.22 GEN1_DEC—LPC I/F Generic Decode Range 1 Register (LPC I/F—D31:F0) ..... 369
13.1.23 GEN2_DEC—LPC I/F Generic Decode Range 2Register (LPC I/F—D31:F0) ..... 369
13.1.24 GEN3_DEC—LPC I/F Generic Decode Range 3Register (LPC I/F—D31:F0) ..... 370
13.1.25 GEN4_DEC—LPC I/F Generic Decode Range 4Register (LPC I/F—D31:F0) ..... 370
13.1.26 FWH_SEL1—Firmware Hub Select 1 Register (LPC I/F—D31:F0) ..... 371
13.1.27 FWH_SEL2—Firmware Hub Select 2 Register (LPC I/F—D31:F0) ..... 372
13.1.28 FWH_DEC_EN1—Firmware Hub Decode Enable Register (LPC I/F—D31:F0) ..... 372
13.1.29 BIOS_CNTL—BIOS Control Register (LPC I/F—D31:F0) ..... 374
13.1.30 FDCAP—Feature Detection Capability ID (LPC I/F—D31:F0) ..... 375
13.1.31 FDLEN—Feature Detection Capability Length (LPC I/F—D31:F0) ..... 375

13.1.32 FDVER—Feature Detection Version (LPC I/F—D31:F0) ..... 376
13.1.33 FDVCT—Feature Vector Register (LPC I/F—D31:F0) ..... 376
13.1.34 RCBA—Root Complex Base Address Register (LPC I/F—D31:F0) ..... 376
13.2 DMA I/O Registers (LPC I/F—D31:F0) ..... 377
13.2.1 DMABASE_CA—DMA Base and Current Address Registers (LPC I/F—D31:F0) ..... 378
13.2.2 DMABASE_CC—DMA Base and Current Count Registers (LPC I/F—D31:F0) ..... 378
13.2.3 DMAMEM_LP—DMA Memory Low Page Registers (LPC I/F—D31:F0) ..... 379
13.2.4 DMACMD—DMA Command Register (LPC I/F—D31:F0) ..... 379
13.2.5 DMASTA—DMA Status Register (LPC I/F—D31:F0) ..... 380
13.2.6 DMA_WRSMSK—DMA Write Single Mask Register (LPC I/F—D31:F0) ..... 380
13.2.7 DMACH_MODE—DMA Channel Mode Register (LPC I/F—D31:F0) ..... 381
13.2.8 DMA Clear Byte Pointer Register (LPC I/F—D31:F0) ..... 382
13.2.9 DMA Master Clear Register (LPC I/F—D31:F0) ..... 382
13.2.10 DMA_CLMSK—DMA Clear Mask Register (LPC I/F—D31:F0) ..... 382
13.2.11 DMA_WRMSK—DMA Write All Mask Register (LPC I/F—D31:F0) ..... 383
13.3 Timer I/O Registers (LPC I/F—D31:F0) ..... 383
13.3.1 TCW—Timer Control Word Register (LPC I/F—D31:F0) ..... 384
13.3.2 RDBK_CMD—Read Back Command (LPC I/F—D31:F0) ..... 385
13.3.3 LTCH_CMD—Counter Latch Command (LPC I/F—D31:F0) ..... 385
13.3.4 SBYTE_FMT—Interval Timer Status Byte Format Register (LPC I/F—D31:F0) ..... 386
13.3.5 Counter Access Ports Register (LPC I/F—D31:F0) ..... 387
13.4 8259 Interrupt Controller (PIC) Registers (LPC I/F—D31:F0) ..... 387
13.4.1 Interrupt Controller I/O MAP (LPC I/F—D31:F0) ..... 387
13.4.2 ICW1—Initialization Command Word 1 Register (LPC I/F—D31:F0) ..... 388
13.4.3 ICW2—Initialization Command Word 2 Register (LPC I/F—D31:F0) ..... 389
13.4.4 ICW3—Master Controller Initialization Command Word 3 Register (LPC I/F—D31:F0) ..... 390
13.4.5 ICW3—Slave Controller Initialization Command Word 3 Register (LPC I/F—D31:F0) ..... 390
13.4.6 ICW4—Initialization Command Word 4 Register (LPC I/F—D31:F0) ..... 390
13.4.7 OCW1—Operational Control Word 1 (Interrupt Mask) Register (LPC I/F—D31:F0) ..... 391
13.4.8 OCW2—Operational Control Word 2 Register (LPC I/F—D31:F0) ..... 391
13.4.9 OCW3—Operational Control Word 3 Register (LPC I/F—D31:F0) ..... 392
13.4.10 ELCR1—Master Controller Edge/Level Triggered Register (LPC I/F—D31:F0) ..... 393
13.4.11 ELCR2—Slave Controller Edge/Level Triggered Register (LPC I/F—D31:F0) ..... 393
13.5 Advanced Programmable Interrupt Controller (APIC)(D31:F0) ..... 394
13.5.1 APIC Register Map (LPC I/F—D31:F0) ..... 394

13.5.2 IND—Index Register (LPC I/F—D31:F0) ..... 395
13.5.3 DAT—Data Register (LPC I/F—D31:F0) ..... 395
13.5.4 EOIR—EOI Register (LPC I/F—D31:F0) ..... 395
13.5.5 ID—Identification Register (LPC I/F—D31:F0) ..... 396
13.5.6 VER—Version Register (LPC I/F—D31:F0) ..... 396
13.5.7 REDIR_TBL—Redirection Table (LPC I/F—D31:F0) ..... 397
13.6 Real Time Clock Registers (LPC I/F—D31:F0) ..... 399
13.6.1 I/O Register Address Map (LPC I/F—D31:F0) ..... 399
13.6.2 Indexed Registers (LPC I/F—D31:F0) ..... 399
13.7 Processor Interface Registers (LPC I/F—D31:F0) ..... 403
13.7.1 NMI_SC—NMI Status and Control Register (LPC I/F—D31:F0) ..... 404
13.7.2 NMI_EN—NMI Enable (and Real Time Clock Index) Register (LPC I/F—D31:F0) ..... 405
13.7.3 PORT92—Fast A20 and Init Register (LPC I/F—D31:F0) ..... 405
13.7.4 COPROC_ERR—Coprocessor Error Register (LPC I/F—D31:F0) ..... 405
13.7.5 RST_CNT—Reset Control Register (LPC I/F—D31:F0) ..... 406
13.8 Power Management Registers (PM—D31:F0) ..... 406
13.8.1 Power Management PCI Configuration Registers (PM—D31:F0) ..... 406
13.8.2 APM I/O Decode ..... 415
13.8.3 Power Management I/O Registers ..... 415
13.9 System Management TCO Registers (D31:F0) ..... 438
13.9.1 TCO_RLD—TCO Timer Reload and Current Value Register ..... 439
13.9.2 TCO_DAT_IN—TCO Data In Register ..... 439
13.9.3 TCO_DAT_OUT—TCO Data Out Register ..... 440
13.9.4 TCO1_STS—TCO1 Status Register ..... 440
13.9.5 TCO2_STS—TCO2 Status Register ..... 442
13.9.6 TCO1_CNT—TCO1 Control Register ..... 443
13.9.7 TCO2_CNT—TCO2 Control Register ..... 443
13.9.8 TCO_MESSAGE1 and TCO_MESSAGE2 Registers ..... 444
13.9.9 TCO_WDCNT—TCO Watchdog Control Register ..... 444
13.9.10 SW_IRQ_GEN—Software IRQ Generation Register ..... 445
13.9.11 TCO_TMR—TCO Timer Initial Value Register ..... 445
13.10 General Purpose I/O Registers (D31:F0) ..... 445
13.10.1 GPIO_USE_SEL—GPIO Use Select Register ..... 446
13.10.2 GP_IO_SEL—GPIO Input/Output Select Register ..... 447
13.10.3 GP_LVL—GPIO Level for Input or Output Register ..... 447
13.10.4 GPO_BLINK—GPO Blink Enable Register ..... 447
13.10.5 GPI_INV—GPIO Signal Invert Register ..... 448
13.10.6 GPIO_USE_SEL2—GPIO Use Select 2 Register[63:32] ..... 448
13.10.7 GP_IO_SEL2—GPIO Input/Output Select 2 Register[63:32] ..... 449
13.10.8 GP_LVL2—GPIO Level for Input or Output 2 Register[63:32] ..... 449

**14 UHCI Controllers Registers** ..... 450

14.1 PCI Configuration Registers (USB—D29:F0/F1/F2/F3) ..... 450
14.1.1 VID—Vendor Identification Register (USB—D29:F0/F1/F2/F3) ..... 451
14.1.2 DID—Device Identification Register (USB—D29:F0/F1/F2/F3) ..... 451
14.1.3 PCICMD—PCI Command Register (USB—D29:F0/F1/F2/F3) ..... 451

14.1.4 PCISTS—PCI Status Register (USB—D29:F0/F1/F2/F3) ..... 452
14.1.5 RID—Revision Identification Register (USB—D29:F0/F1/F2/F3) ..... 453
14.1.6 PI—Programming Interface Register (USB—D29:F0/F1/F2/F3) ..... 453
14.1.7 SCC—Sub Class Code Register (USB—D29:F0/F1/F2/F3) ..... 453
14.1.8 BCC—Base Class Code Register (USB—D29:F0/F1/F2/F3) ..... 453
14.1.9 MLT—Master Latency Timer Register (USB—D29:F0/F1/F2/F3) ..... 454
14.1.10 HEADTYP—Header Type Register (USB—D29:F0/F1/F2/F3) ..... 454
14.1.11 BASE—Base Address Register (USB—D29:F0/F1/F2/F3) ..... 455
14.1.12 SVID — Subsystem Vendor Identification Register (USB—D29:F0/F1/F2/F3) ..... 455
14.1.13 SID — Subsystem Identification Register (USB—D29:F0/F1/F2/F3) ..... 455
14.1.14 INT_LN—Interrupt Line Register (USB—D29:F0/F1/F2/F3) ..... 456
14.1.15 INT_PN—Interrupt Pin Register (USB—D29:F0/F1/F2/F3) ..... 456
14.1.16 USB_RELNUM—Serial Bus Release Number Register (USB—D29:F0/F1/F2/F3) ..... 456
14.1.17 USB_LEGKEY—USB Legacy Keyboard/Mouse Control Register (USB—D29:F0/F1/F2/F3) ..... 457
14.1.18 USB_RES—USB Resume Enable Register (USB—D29:F0/F1/F2/F3) ..... 459
14.1.19 CWP—Core Well Policy Register (USB—D29:F0/F1/F2/F3) ..... 459
14.2 USB I/O Registers ..... 459
14.2.1 USBCMD—USB Command Register ..... 460
14.2.2 USBSTS—USB Status Register ..... 463
14.2.3 USBINTR—USB Interrupt Enable Register ..... 464
14.2.4 FRNUM—Frame Number Register ..... 465
14.2.5 FRBASEADD—Frame List Base Address Register ..... 465
14.2.6 SOFMOD—Start of Frame Modify Register ..... 466
14.2.7 PORTSC[0,1]—Port Status and Control Register ..... 467

**15 SATA Controller Registers (D31:F2)** ..... 470

15.1 PCI Configuration Registers (SATA–D31:F2) ..... 470
15.1.1 VID—Vendor Identification Register (SATA—D31:F2) ..... 472
15.1.2 DID—Device Identification Register (SATA—D31:F2) ..... 472
15.1.3 PCICMD—PCI Command Register (SATA–D31:F2) ..... 472
15.1.4 PCISTS — PCI Status Register (SATA–D31:F2) ..... 473
15.1.5 RID—Revision Identification Register (SATA—D31:F2) ..... 474
15.1.6 PI—Programming Interface Register (SATA–D31:F2) ..... 474
15.1.7 SCC—Sub Class Code Register (SATA–D31:F2) ..... 475
15.1.8 BCC—Base Class Code Register (SATA–D31:F2SATA–D31:F2) ..... 476
15.1.9 PMLT—Primary Master Latency Timer Register (SATA–D31:F2) ..... 476

15.1.10 PCMD_BAR—Primary Command Block Base Address Register (SATA–D31:F2) ..... 476
15.1.11 PCNL_BAR—Primary Control Block Base Address Register (SATA–D31:F2) ..... 476
15.1.12 SCMD_BAR—Secondary Command Block Base Address Register (IDE D31:F1) ..... 477
15.1.13 SCNL_BAR—Secondary Control Block Base Address Register (IDE D31:F1) ..... 477
15.1.14 BAR — Legacy Bus Master Base Address Register (SATA–D31:F2) ..... 477
15.1.15 ABAR — AHCI Base Address Register (SATA–D31:F2) ..... 478
15.1.16 SVID—Subsystem Vendor Identification Register (SATA–D31:F2) ..... 479
15.1.17 SID—Subsystem Identification Register (SATA–D31:F2) ..... 479
15.1.18 CAP—Capabilities Pointer Register (SATA–D31:F2) ..... 479
15.1.19 INT_LN—Interrupt Line Register (SATA–D31:F2) ..... 479
15.1.20 INT_PN—Interrupt Pin Register (SATA–D31:F2) ..... 480
15.1.21 IDE_TIMP — Primary IDE Timing Register (SATA–D31:F2) ..... 480
15.1.22 IDE_TIMS — Slave IDE Timing Register (SATA–D31:F2) ..... 481
15.1.23 SDMA_CNT—Synchronous DMA Control Register (SATA–D31:F2) ..... 482
15.1.24 SDMA_TIM—Synchronous DMA Timing Register (SATA–D31:F2) ..... 483
15.1.25 IDE_CONFIG—IDE I/O Configuration Register (SATA–D31:F2) ..... 485
15.1.26 PID—PCI Power Management Capability Identification Register (SATA–D31:F2) ..... 486
15.1.27 PC—PCI Power Management Capabilities Register (SATA–D31:F2) ..... 486
15.1.28 PMCS—PCI Power Management Control and Status Register (SATA–D31:F2) ..... 487
15.1.29 MSICI—Message Signaled Interrupt Capability Identification (SATA–D31:F2) 487
15.1.30 MSIMC—Message Signaled Interrupt Message Control (SATA–D31:F2) ..... 488
15.1.31 MSIMA— Message Signaled Interrupt Message Address (SATA–D31:F2) ..... 489
15.1.32 MSIMD—Message Signaled Interrupt Message Data (SATA–D31:F2) ..... 489
15.1.33 MAP—Address Map Register (SATA–D31:F2) ..... 489
15.1.34 PCS—Port Control and Status Register (SATA–D31:F2) ..... 490
15.1.35 SIR—SATA Initialization Register ..... 491
15.1.36 SIRI—SATA Indexed Registers Index ..... 492
15.1.37 STRD—SATA Indexed Register Data ..... 492
15.1.38 SCAP0—SATA Capability Register 0 (SATA–D31:F2) ..... 493
15.1.39 SCAP1—SATA Capability Register 1 (SATA–D31:F2) ..... 494
15.1.40 ATC—APM Trapping Control Register (SATA–D31:F2) ..... 495
15.1.41 ATS—APM Trapping Status Register (SATA–D31:F2) ..... 495
15.1.42 SP — Scratch Pad Register (SATA–D31:F2) ..... 495
15.1.43 BFCS—BIST FIS Control/Status Register (SATA–D31:F2) ..... 496
15.1.44 BFTD1—BIST FIS Transmit Data1 Register (SATA–D31:F2) ..... 497
15.1.45 BFTD2—BIST FIS Transmit Data2 Register (SATA–D31:F2) ..... 497
15.2 Bus Master IDE I/O Registers (D31:F2) ..... 497
15.2.1 BMIC[P,S]—Bus Master IDE Command Register (D31:F2) ..... 498
15.2.2 BMIS[P,S]—Bus Master IDE Status Register (D31:F2) ..... 499
15.2.3 BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register (D31:F2) ..... 500

15.3 AHCI Registers (D31:F2) ..... 501
15.3.1 AHCI Generic Host Control Registers (D31:F2) ..... 501
15.3.2 Port Registers (D31:F2) ..... 505

**16 EHCI Controller Registers (D29:F7)** ..... 519

16.1 USB EHCI Configuration Registers (USB EHCI—D29:F7) ..... 519
16.1.1 VID—Vendor Identification Register (USB EHCI—D29:F7) ..... 520
16.1.2 DID—Device Identification Register (USB EHCI—D29:F7) ..... 520
16.1.3 PCICMD—PCI Command Register (USB EHCI—D29:F7) ..... 521
16.1.4 PCISTS—PCI Status Register (USB EHCI—D29:F7) ..... 522
16.1.5 RID—Revision Identification Register (USB EHCI—D29:F7) ..... 523
16.1.6 PI—Programming Interface Register (USB EHCI—D29:F7) ..... 523
16.1.7 SCC—Sub Class Code Register (USB EHCI—D29:F7) ..... 523
16.1.8 BCC—Base Class Code Register (USB EHCI—D29:F7) ..... 523
16.1.9 PMLT—Primary Master Latency Timer Register (USB EHCI—D29:F7) ..... 524
16.1.10 MEM_BASE—Memory Base Address Register (USB EHCI—D29:F7) ..... 524
16.1.11 SVID—USB EHCI Subsystem Vendor ID Register (USB EHCI—D29:F7) ..... 524
16.1.12 SID—USB EHCI Subsystem ID Register (USB EHCI—D29:F7) ..... 525
16.1.13 CAP_PTR—Capabilities Pointer Register (USB EHCI—D29:F7) ..... 525
16.1.14 INT_LN—Interrupt Line Register (USB EHCI—D29:F7) ..... 525
16.1.15 INT_PN—Interrupt Pin Register (USB EHCI—D29:F7) ..... 525
16.1.16 PWR_CAPID—PCI Power Management Capability ID Register (USB EHCI—D29:F7) ..... 525
16.1.17 NXT_PTR1—Next Item Pointer #1 Register (USB EHCI—D29:F7) ..... 526
16.1.18 PWR_CAP—Power Management Capabilities Register (USB EHCI—D29:F7) ..... 526
16.1.19 PWR_CNTL_STS—Power Management Control/ Status Register (USB EHCI—D29:F7) ..... 527
16.1.20 DEBUG_CAPID—Debug Port Capability ID Register (USB EHCI—D29:F7) ..... 528
16.1.21 NXT_PTR2—Next Item Pointer #2 Register (USB EHCI—D29:F7) ..... 528
16.1.22 DEBUG_BASE—Debug Port Base Offset Register (USB EHCI—D29:F7) ..... 528
16.1.23 USB_RELNUM—USB Release Number Register (USB EHCI—D29:F7) ..... 528
16.1.24 FL_ADJ—Frame Length Adjustment Register (USB EHCI—D29:F7) ..... 529

16.1.25 PWAKE_CAP—Port Wake Capability Register (USB EHCI—D29:F7) ..... 530
16.1.26 LEG_EXT_CAP—USB EHCI Legacy Support Extended Capability Register (USB EHCI—D29:F7) ..... 530
16.1.27 LEG_EXT_CS—USB EHCI Legacy Support Extended Control / Status Register (USB EHCI—D29:F7) ..... 531
16.1.28 SPECIAL_SMI—Intel Specific USB 2.0 SMI Register (USB EHCI—D29:F7) ..... 533
16.1.29 ACCESS_CNTL—Access Control Register (USB EHCI—D29:F7) ..... 534
16.2 Memory-Mapped I/O Registers ..... 534
16.2.1 Host Controller Capability Registers ..... 535
16.2.2 Host Controller Operational Registers ..... 537
16.2.3 USB 2.0-Based Debug Port Register ..... 551

**17 SMBus Controller Registers (D31:F3) ..... 555**

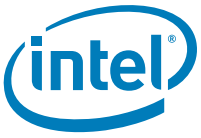
17.1 PCI Configuration Registers (SMBUS—D31:F3) ..... 555
17.1.1 VID—Vendor Identification Register (SMBUS—D31:F3) ..... 555
17.1.2 DID—Device Identification Register (SMBUS—D31:F3) ..... 556
17.1.3 PCICMD—PCI Command Register (SMBUS—D31:F3) ..... 556
17.1.4 PCISTS—PCI Status Register (SMBUS—D31:F3) ..... 557
17.1.5 RID—Revision Identification Register (SMBUS—D31:F3) ..... 557
17.1.6 PI—Programming Interface Register (SMBUS—D31:F3) ..... 558
17.1.7 SCC—Sub Class Code Register (SMBUS—D31:F3) ..... 558
17.1.8 BCC—Base Class Code Register (SMBUS—D31:F3) ..... 558
17.1.9 SMB_BASE—SMBUS Base Address Register (SMBUS—D31:F3) ..... 558
17.1.10 SVID — Subsystem Vendor Identification Register (SMBUS—D31:F2/F4) ..... 559
17.1.11 SID — Subsystem Identification Register (SMBUS—D31:F2/F4) ..... 559
17.1.12 INT_LN—Interrupt Line Register (SMBUS—D31:F3) ..... 559
17.1.13 INT_PN—Interrupt Pin Register (SMBUS—D31:F3) ..... 559
17.1.14 HOSTC—Host Configuration Register (SMBUS—D31:F3) ..... 560
17.2 SMBus I/O Registers ..... 560
17.2.1 HST_STS—Host Status Register (SMBUS—D31:F3) ..... 561
17.2.2 HST_CNT—Host Control Register (SMBUS—D31:F3) ..... 563
17.2.3 HST_CMD—Host Command Register (SMBUS—D31:F3) ..... 565
17.2.4 XMIT_SLVA—Transmit Slave Address Register (SMBUS—D31:F3) ..... 565
17.2.5 HST_D0—Host Data 0 Register (SMBUS—D31:F3) ..... 565
17.2.6 HST_D1—Host Data 1 Register (SMBUS—D31:F3) ..... 565
17.2.7 Host_BLOCK_DB—Host Block Data Byte Register (SMBUS—D31:F3) ..... 566
17.2.8 PEC—Packet Error Check (PEC) Register (SMBUS—D31:F3) ..... 566
17.2.9 RCV_SLVA—Receive Slave Address Register (SMBUS—D31:F3) ..... 567
17.2.10 SLV_DATA—Receive Slave Data Register (SMBUS—D31:F3) ..... 567
17.2.11 AUX_STS—Auxiliary Status Register (SMBUS—D31:F3) ..... 567
17.2.12 AUX_CTL—Auxiliary Control Register (SMBUS—D31:F3) ..... 568
17.2.13 SMLINK_PIN_CTL—SMLink Pin Control Register (SMBUS—D31:F3) ..... 568

17.2.14 SMBUS_PIN_CTL—SMBUS Pin Control Register (SMBUS—D31:F3) ..... 569
17.2.15 SLV_STS—Slave Status Register (SMBUS—D31:F3) ..... 570
17.2.16 SLV_CMD—Slave Command Register (SMBUS—D31:F3) ..... 570
17.2.17 NOTIFY_DADDR—Notify Device Address Register (SMBUS—D31:F3) ..... 571
17.2.18 NOTIFY_DLOW—Notify Data Low Byte Register (SMBUS—D31:F3) ..... 571
17.2.19 NOTIFY_DHIGH—Notify Data High Byte Register (SMBUS—D31:F3) ..... 571

**18** Intel HD Audio Controller Registers (D27:F0) ..... 573

18.1 Intel HD Audio PCI Configuration Space (Intel HD Audio— D27:F0) ..... 573
18.1.1 VID—Vendor Identification Register (Intel HD Audio Controller—D27:F0) ..... 575
18.1.2 DID—Device Identification Register (Intel HD Audio Controller—D27:F0) ..... 575
18.1.3 PCICMD—PCI Command Register (Intel HD Audio Controller—D27:F0) ..... 575
18.1.4 PCISTS—PCI Status Register (Intel HD Audio Controller—D27:F0) ..... 576
18.1.5 RID—Revision Identification Register (Intel HD Audio Controller—D27:F0) ..... 577
18.1.6 PI—Programming Interface Register (Intel HD Audio Controller—D27:F0) ..... 577
18.1.7 SCC—Sub Class Code Register (Intel HD Audio Controller—D27:F0) ..... 577
18.1.8 BCC—Base Class Code Register (Intel HD Audio Controller—D27:F0) ..... 577
18.1.9 CLS—Cache Line Size Register (Intel HD Audio Controller—D27:F0) ..... 577
18.1.10 LT—Latency Timer Register (Intel HD Audio Controller—D27:F0) ..... 578
18.1.11 HEADTYP—Header Type Register (Intel HD Audio Controller—D27:F0) ..... 578
18.1.12 HDBARL—Intel HD Audio Lower Base Address Register (Intel HD Audio—D27:F0) ..... 578
18.1.13 HDBARU—Intel HD Audio Upper Base Address Register (Intel HD Audio Controller—D27:F0) ..... 578
18.1.14 SVID—Subsystem Vendor Identification Register (Intel HD Audio Controller—D27:F0) ..... 579
18.1.15 SID—Subsystem Identification Register (Intel HD Audio Controller—D27:F0) ..... 579
18.1.16 CAPPTR—Capabilities Pointer Register (Audio—D30:F2) ..... 579
18.1.17 INTLN—Interrupt Line Register (Intel HD Audio Controller—D27:F0) ..... 580
18.1.18 INTPN—Interrupt Pin Register (Intel HD Audio Controller—D27:F0) ..... 580
18.1.19 HDCTL—Intel HD Audio Control Register (Intel HD Audio Controller—D27:F0) ..... 580
18.1.20 TCSEL—Traffic Class Select Register (Intel HD Audio Controller—D27:F0) ..... 581
18.1.21 PID—PCI Power Management Capability ID Register (Intel HD Audio Controller—D27:F0) ..... 582

18.1.22 PC—Power Management Capabilities Register (Intel HD Audio Controller—D27:F0) ..... 582
18.1.23 PCS—Power Management Control and Status Register (Intel HD Audio Controller—D27:F0) ..... 582
18.1.24 MID—MSI Capability ID Register (Intel HD Audio Controller—D27:F0) ..... 583
18.1.25 MMC—MSI Message Control Register (Intel HD Audio Controller—D27:F0) ..... 584
18.1.26 MMLA—MSI Message Lower Address Register (Intel HD Audio Controller—D27:F0) ..... 584
18.1.27 MMUA—MSI Message Upper Address Register (Intel HD Audio Controller—D27:F0) ..... 584
18.1.28 MMD—MSI Message Data Register (Intel HD Audio Controller—D27:F0) ..... 584
18.1.29 PXID—PCI Express* Capability ID Register (Intel HD Audio Controller—D27:F0) ..... 585
18.1.30 PXC—PCI Express* Capabilities Register (Intel HD Audio Controller—D27:F0) ..... 585
18.1.31 DEVCAP—Device Capabilities Register (Intel HD Audio Controller—D27:F0) ..... 585
18.1.32 DEVC—Device Control Register (Intel HD Audio Controller—D27:F0) ..... 586
18.1.33 DEVS—Device Status Register (Intel HD Audio Controller—D27:F0) ..... 586
18.1.34 VCCAP—Virtual Channel Enhanced Capability Header (Intel HD Audio Controller—D27:F0) ..... 587
18.1.35 PVCCAP1—Port VC Capability Register 1 (Intel HD Audio Controller—D27:F0) ..... 587
18.1.36 PVCCAP2 — Port VC Capability Register 2 (Intel HD Audio Controller—D27:F0) ..... 587
18.1.37 PVCCTL — Port VC Control Register (Intel HD Audio Controller—D27:F0) ..... 588
18.1.38 PVCSTS—Port VC Status Register (Intel HD Audio Controller—D27:F0) ..... 588
18.1.39 VC0CAP—VC0 Resource Capability Register (Intel HD Audio Controller—D27:F0) ..... 588
18.1.40 VC0CTL—VC0 Resource Control Register (Intel HD Audio Controller—D27:F0) ..... 589
18.1.41 VC0STS—VC0 Resource Status Register (Intel HD Audio Controller—D27:F0) ..... 589
18.1.42 VCiCAP—VCi Resource Capability Register (Intel HD Audio Controller—D27:F0) ..... 589
18.1.43 VCiCTL—VCi Resource Control Register (Intel HD Audio Controller—D27:F0) ..... 590
18.1.44 VCiSTS—VCi Resource Status Register (Intel HD Audio Controller—D27:F0) ..... 590
18.1.45 RCCAP—Root Complex Link Declaration Enhanced Capability Header Register (Intel HD Audio Controller—D27:F0) ..... 591
18.1.46 ESD—Element Self Description Register (Intel HD Audio Controller—D27:F0) ..... 591
18.1.47 L1DESC—Link 1 Description Register (Intel HD Audio Controller—D27:F0) ..... 591
18.1.48 L1ADDL—Link 1 Lower Address Register (Intel HD Audio Controller—D27:F0) ..... 592
18.1.49 L1ADDU—Link 1 Upper Address Register (Intel HD Audio Controller—D27:F0) ..... 592

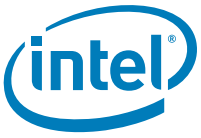



18.2 Intel HD Audio Memory-Mapped Configuration Registers (Intel HD Audio— D27:F0) ..... 592
18.2.1 GCAP—Global Capabilities Register (Intel HD Audio Controller—D27:F0) ..... 596
18.2.2 VMIN—Minor Version Register (Intel HD Audio Controller—D27:F0) ..... 597
18.2.3 VMAJ—Major Version Register (Intel HD Audio Controller—D27:F0) ..... 597
18.2.4 OUTPAY—Output Payload Capability Register (Intel HD Audio Controller—D27:F0) ..... 597
18.2.5 INPAY—Input Payload Capability Register (Intel HD Audio Controller—D27:F0) ..... 598
18.2.6 GCTL—Global Control Register (Intel HD Audio Controller—D27:F0) ..... 598
18.2.7 WAKEEN—Wake Enable Register (Intel HD Audio Controller—D27:F0) ..... 599
18.2.8 STATESTS—State Change Status Register (Intel HD Audio Controller—D27:F0) ..... 600
18.2.9 GSTS—Global Status Register (Intel HD Audio Controller—D27:F0) ..... 600
18.2.10 OUTSTRMPAY—Output Stream Payload Capability (Intel HD Audio Controller—D27:F0) ..... 600
18.2.11 INSTRMPAY—Input Stream Payload Capability (Intel HD Audio Controller—D27:F0) ..... 601
18.2.12 INTCTL—Interrupt Control Register (Intel HD Audio Controller—D27:F0) ..... 602
18.2.13 INTSTS—Interrupt Status Register (Intel HD Audio Controller—D27:F0) ..... 603
18.2.14 WALCLK—Wall Clock Counter Register (Intel HD Audio Controller—D27:F0) ..... 604
18.2.15 SSYNC—Stream Synchronization Register (Intel HD Audio Controller—D27:F0) ..... 604
18.2.16 CORBLBASE—CORB Lower Base Address Register (Intel HD Audio Controller—D27:F0) ..... 605
18.2.17 CORBUBASE—CORB Upper Base Address Register (Intel HD Audio Controller—D27:F0) ..... 605
18.2.18 CORBWP—CORB Write Pointer Register (Intel HD Audio Controller—D27:F0) ..... 605
18.2.19 CORBRP—CORB Read Pointer Register (Intel HD Audio Controller—D27:F0) ..... 606
18.2.20 CORBCTL—CORB Control Register (Intel HD Audio Controller—D27:F0) ..... 606
18.2.21 CORBST—CORB Status Register (Intel HD Audio Controller—D27:F0) ..... 607
18.2.22 CORBSIZE—CORB Size Register Intel HD Audio Controller—D27:F0) ..... 607
18.2.23 RIRBLBASE—RIRB Lower Base Address Register (Intel HD Audio Controller—D27:F0) ..... 607
18.2.24 RIRBUBASE—RIRB Upper Base Address Register (Intel HD Audio Controller—D27:F0) ..... 608
18.2.25 RIRBWP—RIRB Write Pointer Register (Intel HD Audio Controller—D27:F0) ..... 608
18.2.26 RINTCNT—Response Interrupt Count Register (Intel HD Audio Controller—D27:F0) ..... 608
18.2.27 RIRBCTL—RIRB Control Register (Intel HD Audio Controller—D27:F0) ..... 609

18.2.28 RIRBSTS—RIRB Status Register (Intel HD Audio Controller—D27:F0) ..... 609
18.2.29 RIRBSIZE—RIRB Size Register (Intel HD Audio Controller—D27:F0) ..... 610
18.2.30 IC—Immediate Command Register (Intel HD Audio Controller—D27:F0) ..... 610
18.2.31 IR—Immediate Response Register (Intel HD Audio Controller—D27:F0) ..... 611
18.2.32 IRS—Immediate Command Status Register (Intel HD Audio Controller—D27:F0) ..... 611
18.2.33 DPLBASE—DMA Position Lower Base Address Register (Intel HD Audio Controller—D27:F0) ..... 612
18.2.34 DPUBASE—DMA Position Upper Base Address Register (Intel HD Audio Controller—D27:F0) ..... 612
18.2.35 SDCTL—Stream Descriptor Control Register (Intel HD Audio Controller—D27:F0) ..... 612
18.2.36 SDSTS—Stream Descriptor Status Register (Intel HD Audio Controller—D27:F0) ..... 614
18.2.37 SDLPIB—Stream Descriptor Link Position in Buffer Register (Intel HD Audio Controller—D27:F0) ..... 615
18.2.38 SDCBL—Stream Descriptor Cyclic Buffer Length Register (Intel HD Audio Controller—D27:F0) ..... 615
18.2.39 SDLVI—Stream Descriptor Last Valid Index Register (Intel HD Audio Controller—D27:F0) ..... 616
18.2.40 SDFIFOW—Stream Descriptor FIFO Watermark Register (Intel HD Audio Controller—D27:F0) ..... 616
18.2.41 SDFIFOS—Stream Descriptor FIFO Size Register (Intel HD Audio Controller—D27:F0) ..... 617
18.2.42 SDFMT—Stream Descriptor Format Register (Intel HD Audio Controller—D27:F0) ..... 618
18.2.43 SDBDPL—Stream Descriptor Buffer Descriptor List Pointer Lower Base Address Register (Intel HD Audio Controller—D27:F0) ..... 619
18.2.44 SDBDPU—Stream Descriptor Buffer Descriptor List Pointer Upper Base Address Register (Intel HD Audio Controller—D27:F0) ..... 620

**19 PCI Express* Configuration Registers** ..... 621

19.1 PCI Express* Configuration Registers (PCI Express—D28:F0/F1/F2/F3) ..... 621
19.1.1 VID—Vendor Identification Register (PCI Express—D28:F0/F1/F2/F3) ..... 623
19.1.2 DID—Device Identification Register (PCI Express—D28:F0/F1/F2/F3) ..... 623
19.1.3 PCICMD—PCI Command Register (PCI Express—D28:F0/F1/F2/F3) ..... 623
19.1.4 PCISTS—PCI Status Register (PCI Express—D28:F0/F1/F2/F3) ..... 624
19.1.5 RID—Revision Identification Register (PCI Express—D28:F0/F1/F2/F3) ..... 625
19.1.6 PI—Programming Interface Register (PCI Express—D28:F0/F1/F2/F3) ..... 625
19.1.7 SCC—Sub Class Code Register (PCI Express—D28:F0/F1/F2/F3) ..... 626
19.1.8 BCC—Base Class Code Register (PCI Express—D28:F0/F1/F2/F3) ..... 626



19.1.9 CLS—Cache Line Size Register (PCI Express—D28:F0/F1/F2/F3) ..... 626
19.1.10 PLT—Primary Latency Timer Register (PCI Express—D28:F0/F1/F2/F3) ..... 626
19.1.11 HEADTYP—Header Type Register (PCI Express—D28:F0/F1/F2/F3) ..... 627
19.1.12 BNUM—Bus Number Register (PCI Express—D28:F0/F1/F2/F3) ..... 627
19.1.13 IOBL—I/O Base and Limit Register (PCI Express—D28:F0/F1/F2/F3) ..... 627
19.1.14 SSTS—Secondary Status Register (PCI Express—D28:F0/F1/F2/F3) ..... 628
19.1.15 MBL—Memory Base and Limit Register (PCI Express—D28:F0/F1/F2/F3) ..... 628
19.1.16 PMBL—Prefetchable Memory Base and Limit Register (PCI Express—D28:F0/F1/F2/F3) ..... 629
19.1.17 PMBU32—Prefetchable Memory Base Upper 32 Bits Register (PCI Express—D28:F0/F1/F2/F3) ..... 629
19.1.18 PMLU32—Prefetchable Memory Limit Upper 32 Bits Register (PCI Express—D28:F0/F1/F2/F3) ..... 629
19.1.19 CAPP—Capabilities List Pointer Register (PCI Express—D28:F0/F1/F2/F3) ..... 630
19.1.20 INTR—Interrupt Information Register (PCI Express—D28:F0/F1/F2/F3) ..... 630
19.1.21 BCTRL—Bridge Control Register (PCI Express—D28:F0/F1/F2/F3) ..... 630
19.1.22 CLIST—Capabilities List Register (PCI Express—D28:F0/F1/F2/F3) ..... 631
19.1.23 XCAP—PCI Express* Capabilities Register (PCI Express—D28:F0/F1/F2/F3) ..... 632
19.1.24 DCAP—Device Capabilities Register (PCI Express—D28:F0/F1/F2/F3) ..... 632
19.1.25 DCTL—Device Control Register (PCI Express—D28:F0/F1/F2/F3) ..... 633
19.1.26 DSTS—Device Status Register (PCI Express—D28:F0/F1/F2/F3) ..... 633
19.1.27 LCAP—Link Capabilities Register (PCI Express—D28:F0/F1/F2/F3) ..... 634
19.1.28 LCTL—Link Control Register (PCI Express—D28:F0/F1/F2/F3) ..... 635
19.1.29 LSTS—Link Status Register (PCI Express—D28:F0/F1/F2/F3) ..... 636
19.1.30 SLCAP—Slot Capabilities Register (PCI Express—D28:F0/F1/F2/F3) ..... 637
19.1.31 SLCTL—Slot Control Register (PCI Express—D28:F0/F1/F2/F3) ..... 638
19.1.32 SLSTS—Slot Status Register (PCI Express—D28:F0/F1/F2/F3) ..... 639
19.1.33 RCTL—Root Control Register (PCI Express—D28:F0/F1/F2/F3) ..... 640
19.1.34 RSTS—Root Status Register (PCI Express—D28:F0/F1/F2/F3) ..... 640
19.1.35 MID—Message Signaled Interrupt Identifiers Register (PCI Express—D28:F0/F1/F2/F3) ..... 641
19.1.36 MC—Message Signaled Interrupt Message Control Register (PCI Express—D28:F0/F1/F2/F3) ..... 641

19.1.37 MA—Message Signaled Interrupt Message Address Register (PCI Express—D28:F0/F1/F2/F3) ..... 641
19.1.38 MD—Message Signaled Interrupt Message Data Register (PCI Express—D28:F0/F1/F2/F3) ..... 642
19.1.39 SVCAP—Subsystem Vendor Capability Register (PCI Express—D28:F0/F1/F2/F3) ..... 642
19.1.40 SVID—Subsystem Vendor Identification Register (PCI Express—D28:F0/F1/F2/F3) ..... 642
19.1.41 PMCAP—Power Management Capability Register (PCI Express—D28:F0/F1/F2/F3) ..... 642
19.1.42 PMC—PCI Power Management Capabilities Register (PCI Express—D28:F0/F1/F2/F3) ..... 643
19.1.43 PMCS—PCI Power Management Control and Status Register (PCI Express—D28:F0/F1/F2/F3) ..... 643
19.1.44 MPC—Miscellaneous Port Configuration Register (PCI Express—D28:F0/F1/F2/F3) ..... 644
19.1.45 SMSCS—SMI/SCI Status Register (PCI Express—D28:F0/F1/F2/F3) ..... 646
19.1.46 RPDCGEN - Root Port Dynamic Clock Gating Enable (PCI Express-D28:F0/F1/F2/F3) (Netbook Only) ..... 647
19.1.47 IPWS—Intel® PRO/Wireless 3945ABG Status (PCI Express—D28:F0/F1/F2/F3) (Netbook Only) ..... 647
19.1.48 VCH—Virtual Channel Capability Header Register (PCI Express—D28:F0/F1/F2/F3) ..... 647
19.1.49 VCAP2—Virtual Channel Capability 2 Register (PCI Express—D28:F0/F1/F2/F3) ..... 648
19.1.50 PVC—Port Virtual Channel Control Register (PCI Express—D28:F0/F1/F2/F3) ..... 648
19.1.51 PVS — Port Virtual Channel Status Register (PCI Express—D28:F0/F1/F2/F3) ..... 648
19.1.52 V0CAP — Virtual Channel 0 Resource Capability Register (PCI Express—D28:F0/F1/F2/F3) ..... 649
19.1.53 V0CTL — Virtual Channel 0 Resource Control Register (PCI Express—D28:F0/F1/F2/F3) ..... 649
19.1.54 V0STS — Virtual Channel 0 Resource Status Register (PCI Express—D28:F0/F1/F2/F3) ..... 650
19.1.55 UES — Uncorrectable Error Status Register (PCI Express—D28:F0/F1/F2/F3 ..... 650
19.1.56 UEM — Uncorrectable Error Mask (PCI Express—D28:F0/F1/F2/F3) ..... 651
19.1.57 UEV — Uncorrectable Error Severity (PCI Express—D28:F0/F1/F2/F3) ..... 652
19.1.58 CES — Correctable Error Status Register (PCI Express—D28:F0/F1/F2/F3) ..... 653
19.1.59 CEM — Correctable Error Mask Register (PCI Express—D28:F0/F1/F2/F3) ..... 654
19.1.60 AECC — Advanced Error Capabilities and Control Register (PCI Express—D28:F0/F1/F2/F3) ..... 654
19.1.61 RES — Root Error Status Register (PCI Express—D28:F0/F1/F2/F3) ..... 655
19.1.62 RCTCL — Root Complex Topology Capability List Register (PCI Express—D28:F0/F1/F2/F3) ..... 655
19.1.63 ESD — Element Self Description Register (PCI Express—D28:F0/F1/F2/F3) ..... 656
19.1.64 ULD — Upstream Link Description Register (PCI Express—D28:F0/F1/F2/F3) ..... 656

19.1.65 ULBA — Upstream Link Base Address Register (PCI Express—D28:F0/F1/F2/F3) ..... 657
19.1.66 PEETM — PCI Express Extended Test Mode Register (PCI Express—D28:F0/F1/F2/F3) ..... 657

**20 High Precision Event Timer Registers** ..... 658

20.1 Memory Mapped Registers ..... 658
20.1.1 GCAP_ID—General Capabilities and Identification Register ..... 659
20.1.2 GEN_CONF—General Configuration Register ..... 660
20.1.3 GINTR_STA—General Interrupt Status Register ..... 660
20.1.4 MAIN_CNT—Main Counter Value Register ..... 661
20.1.5 TIMn_CONF—Timer n Configuration and Capabilities Register ..... 661
20.1.6 TIMn_COMP—Timer n Comparator Value Register ..... 663

**21** Serial Peripheral Interface (SPI) ..... 664

21.1 Serial Peripheral Interface Memory Mapped Configuration Registers ..... 664
21.1.1 SPIS—SPI Status Register (SPI Memory Mapped Configuration Registers) ..... 665
21.1.2 SPIC—SPI Control Register (SPI Memory Mapped Configuration Registers) ..... 666
21.1.3 SPIA—SPI Address Register (SPI Memory Mapped Configuration Registers) ..... 667
21.1.4 SPID[N] —SPI Data N Register (SPI Memory Mapped Configuration Registers) ..... 668
21.1.5 BBAR—BIOS Base Address Register (SPI Memory Mapped Configuration Registers) ..... 669
21.1.6 PREOP—Prefix Opcode Configuration Register (SPI Memory Mapped Configuration Registers) ..... 669
21.1.7 OPTYPE—Opcode Type Configuration Register (SPI Memory Mapped Configuration Registers) ..... 670
21.1.8 OPMENU—Opcode Menu Configuration Register (SPI Memory Mapped Configuration Registers) ..... 670
21.1.9 PBR[N]—Protected BIOS Range [N] (SPI Memory Mapped Configuration Registers) ..... 671


# Figures


Figure 1-1 Intel NM10 Family Express Chipset Features Block Diagram ..... 31
Figure 2-2 Interface Signals Block Diagram ..... 44
Figure 2-3 Example External RTC Circuit ..... 63
Figure 4-4 Nettop Only Conceptual System Clock Diagram ..... 75
Figure 4-5 Netbook Only Conceptual Clock Diagram ..... 75
Figure 5-6 Generation of SERR# to Platform ..... 83
Figure 5-7 64-Word EEPROM Read Instruction Waveform ..... 91
Figure 5-8 LPC Interface Diagram ..... 98
Figure 5-9 LPC Bridge SERR# Generation ..... 104
Figure 5-10 Chipset DMA Controller ..... 104
Figure 5-11 DMA Request Assertion through LDRQ# ..... 108
Figure 5-12 Coprocessor Error Timing Diagram ..... 134
Figure 5-13 SATA Power States ..... 172
Figure 5-14 USB Legacy Keyboard Flow Diagram ..... 183

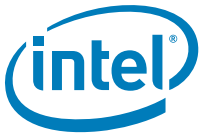



Figure 5-15 Chipset-USB Port Connections ..... 191
Figure 6-16 Chipset Ballout (Topview–Left Side) ..... 219
Figure 6-17 Chipset Ballout (Topview–Right Side) ..... 220
Figure 7-18 Chipset Package (Top View) ..... 225
Figure 7-19 Chipset Package (Bottom View) ..... 226
Figure 7-20 Chipset Package (Side View) ..... 226
Figure 8-21 Clock Timing ..... 244
Figure 8-22 Valid Delay from Rising Clock Edge ..... 244
Figure 8-23 Setup and Hold Times (TBD) ..... 244
Figure 8-24 Float Delay ..... 244
Figure 8-25 Pulse Width ..... 245
Figure 8-26 Output Enable Delay ..... 245
Figure 8-27 USB Rise and Fall Times ..... 245
Figure 8-28 USB Jitter ..... 246
Figure 8-29 USB EOP Width ..... 246
Figure 8-30 SMBus Transaction ..... 246
Figure 8-31 SMBus Timeout ..... 247
Figure 8-32 Power Sequencing and Reset Signal Timings (Nettop Only) ..... 247
Figure 8-33 Power Sequencing and Reset Signal Timings (Netbook Only) ..... 248
Figure 8-34 G3 (Mechanical Off) to S0 Timings (Nettop Only) ..... 249
Figure 8-35 G3 (Mechanical Off) to S0 Timings (Netbook Only) ..... 250
Figure 8-36 S0 to S1 to S0 Timing (Nettop Only) ..... 251
Figure 8-37 S0 to S5 to S0 Timings, $S3_{COLD}$ (Nettop Only) ..... 251
Figure 8-38 S0 to S5 to S0 Timings, $S3_{HOT}$ (Nettop Only) ..... 252
Figure 8-39 S0 to S5 to S0 Timings, $S3_{COLD}$ (Netbook Only) ..... 253
Figure 8-40 S0 to S5 to S0 Timings, $S3_{HOT}$ (Netbook Only) ..... 254
Figure 8-41 C0 to C2 to C0 Timings (Netbook Only) ..... 255
Figure 8-42 C0 to C3 to C0 Timings (Nettop Only) ..... 255
Figure 8-43 C0 to C4 to C0 Timings (Netbook Only) ..... 256
Figure 8-44 Intel HD Audio Input and Output Timings ..... 256
Figure 8-45 SPI Timings ..... 257


## Tables


Table 1-1 Industry Specifications ..... 30
Table 1-2 PCI Devices and Functions ..... 37
Table 2-3 Direct Media Interface Signals ..... 45
Table 2-4 PCI Express* Signals ..... 45
Table 2-5 Platform LAN Connect Interface Signals ..... 45
Table 2-6 EEPROM Interface Signals ..... 46
Table 2-7 Firmware Hub Interface Signals ..... 46
Table 2-8 PCI Interface Signals ..... 47
Table 2-9 Serial ATA Interface Signals ..... 49
Table 2-10 LPC Interface Signals ..... 50
Table 2-11 Interrupt Signals ..... 50
Table 2-12 USB Interface Signals ..... 51
Table 2-13 Power Management Interface Signals ..... 52
Table 2-14 Processor Interface Signals ..... 54
Table 2-15 SM Bus Interface Signals ..... 56
Table 2-16 System Management Interface Signals ..... 56
Table 2-17 Real Time Clock Interface ..... 56

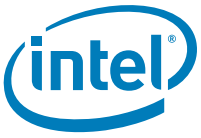
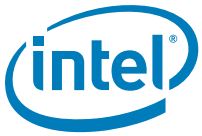



Table 2-18 Other Clocks ..... 57
Table 2-19 Miscellaneous Signals ..... 57
Table 2-20 Intel HD Audio Link Signals ..... 58
Table 2-21 Serial Peripheral Interface (SPI) Signals ..... 59
Table 2-22 General Purpose I/O Signals ..... 59
Table 2-23 Power and Ground Signals ..... 60
Table 2-24 Functional Strap Definitions ..... 61
Table 3-25 Integrated Pull-Up and Pull-Down Resistors ..... 65
Table 3-26 Power Plane and States for Output and I/O Signals ..... 67
Table 3-27 Power Plane for Input Signals ..... 71
Table 4-28 Chipset and System Clock Domains ..... 74
Table 5-29 PCI Bridge Initiator Cycle Types ..... 76
Table 5-30 Type 1 Address Format ..... 79
Table 5-31 MSI vs. PCI IRQ Actions ..... 81
Table 5-32 Advanced TCO Functionality ..... 93
Table 5-33 LPC Cycle Types Supported ..... 99
Table 5-34 Start Field Bit Definitions ..... 99
Table 5-35 Cycle Type Bit Definitions ..... 100
Table 5-36 Transfer Size Bit Definition ..... 100
Table 5-37 SYNC Bit Definition ..... 101
Table 5-38 DMA Transfer Size ..... 106
Table 5-39 Address Shifting in 16-Bit I/O DMA Transfers ..... 106
Table 5-40 Counter Operating Modes ..... 112
Table 5-41 Interrupt Controller Core Connections ..... 115
Table 5-42 Interrupt Status Registers ..... 116
Table 5-43 Content of Interrupt Vector Byte ..... 116
Table 5-44 APIC Interrupt Mapping ..... 123
Table 5-45 Interrupt Message Address Format ..... 125
Table 5-46 Interrupt Message Data Format ..... 125
Table 5-47 Stop Frame Explanation ..... 127
Table 5-48 Data Frame Format ..... 128
Table 5-49 Configuration Bits Reset by RTCRST# Assertion ..... 131
Table 5-50 INIT# Going Active ..... 133
Table 5-51 NMI Sources ..... 135
Table 5-52 DP Signal Differences ..... 135
Table 5-53 General Power States for Systems Using Chipset ..... 137
Table 5-54 State Transition Rules for Chipset ..... 138
Table 5-55 System Power Plane ..... 139
Table 5-56 Causes of SMI# and SCI ..... 140
Table 5-57 Break Events (Netbook) ..... 144
Table 5-58 Sleep Types ..... 148
Table 5-59 Causes of Wake Events ..... 148
Table 5-60 GPI Wake Events ..... 149
Table 5-61 Transitions Due to Power Failure ..... 150
Table 5-62 Transitions Due to Power Button ..... 152
Table 5-63 Transitions Due to RI# Signal ..... 153
Table 5-64 Write Only Registers with Read Paths in ALT Access Mode ..... 156
Table 5-65 PIC Reserved Bits Return Values ..... 158
Table 5-66 Register Write Accesses in ALT Access Mode ..... 158
Table 5-67 Chipset Clock Inputs ..... 161
Table 5-68 Heartbeat Message Data ..... 168
Table 5-69 SATA Features Support in Chipset ..... 169
Table 5-70 SATA Feature Description ..... 169

Table 5-71 Legacy Replacement Routing ..... 175
Table 5-72 Bits Maintained in Low Power States ..... 182
Table 5-73 USB Legacy Keyboard State Transitions ..... 184
Table 5-74 UHCI vs. EHCI ..... 185
Table 5-75 Debug Port Behavior ..... 194
Table 5-76 $I^2C$ Block Read ..... 203
Table 5-77 Enable for SMBALERT# ..... 206
Table 5-78 Enables for SMBus Slave Write and SMBus Host Events ..... 206
Table 5-79 Enables for the Host Notify Command ..... 206
Table 5-80 Slave Write Registers ..... 208
Table 5-81 Command Types ..... 209
Table 5-82 Read Cycle Format ..... 210
Table 5-83 Data Values for Slave Read Registers ..... 210
Table 5-84 Host Notify Format ..... 212
Table 5-85 SPI Implementation Options ..... 214
Table 5-86 Required Commands and Opcodes ..... 215
Table 5-87 Chipset Standard SPI Commands ..... 215
Table 5-88 Flash Protection Mechanism Summary ..... 216
Table 6-89 Chipset Ballout by Signal Name ..... 221
Table 8-90 Chipset Absolute Maximum Ratings ..... 227
Table 8-91 DC Current Characteristics ..... 228
Table 8-92 DC Characteristic Input Signal Association ..... 228
Table 8-93 DC Input Characteristics ..... 229
Table 8-94 DC Characteristic Output Signal Association ..... 231
Table 8-95 DC Output Characteristics ..... 232
Table 8-96 Other DC Characteristics ..... 234
Table 8-97 Clock Timings ..... 235
Table 8-98 SATA Interface Timings ..... 236
Table 8-99 SMBus Timing ..... 236
Table 8-100 Intel HD Audio Timing ..... 237
Table 8-101 LPC Timing ..... 237
Table 8-102 Miscellaneous Timings ..... 238
Table 8-103 SPI Timings ..... 238
Table 8-104 (Power Sequencing and Reset Signal Timings ..... 238
Table 8-105 Power Management Timings ..... 240
Table 9-106 PCI Devices and Functions ..... 259
Table 9-107 Fixed I/O Ranges Decoded by Chipset ..... 260
Table 9-108 Variable I/O Decode Ranges ..... 263
Table 9-109 Memory Decode Ranges from Processor Perspective ..... 264
Table 10-110 Chipset Configuration Register Memory Map (Memory Space) ..... 267
Table 11-111 LAN Controller PCI Register Address Map (LAN Controller—B1:D8:F0) ..... 302
Table 11-112 Configuration of Subsystem ID and Subsystem Vendor ID via EEPROM ..... 309
Table 11-113 Data Register Structure ..... 313
Table 11-114 Chipset Integrated LAN Controller CSR Space Register Address Map ..... 313
Table 11-115 Self-Test Results Format ..... 319
Table 11-116 Statistical Counters ..... 326
Table 11-117 ASF Register Address Map ..... 328
Table 12-118 PCI Bridge Register Address Map (PCI-PCI—D30:F0) ..... 342
Table 13-119 LPC Interface PCI Register Address Map (LPC I/F—D31:F0) ..... 358
Table 13-120 DMA Registers ..... 377
Table 13-121 PIC Registers (LPC I/F—D31:F0) ..... 387
Table 13-122 APIC Direct Registers (LPC I/F—D31:F0) ..... 394
Table 13-123 APIC Indirect Registers (LPC I/F—D31:F0) ..... 394

Table 13-124 RTC I/O Registers (LPC I/F—D31:F0) ..... 399
Table 13-125 RTC (Standard) RAM Bank (LPC I/F—D31:F0) ..... 400
Table 13-126 Processor Interface PCI Register Address Map (LPC I/F—D31:F0) ..... 403
Table 13-127 Power Management PCI Register Address Map (PM—D31:F0) ..... 407
Table 13-128 APM Register Map ..... 415
Table 13-129 ACPI and Legacy I/O Register Map ..... 416
Table 13-130 TCO I/O Register Address Map ..... 438
Table 13-131 Registers to Control GPIO Address Map ..... 445
Table 14-132 UHCI Controller PCI Register Address Map (USB—D29:F0/F1/F2/F3) ..... 450
Table 14-133 USB I/O Registers ..... 460
Table 14-134 Run/Stop, Debug Bit Interaction SWDBG (Bit 5), Run/Stop (Bit 0) Operation 462
Table 15-135 SATA Controller PCI Register Address Map (SATA–D31:F2) ..... 470
Table 15-136 Bus Master IDE I/O Register Address Map ..... 497
Table 15-137 AHCI Register Address Map ..... 501
Table 15-138 Generic Host Controller Register Address Map ..... 501
Table 15-139 Port [1:0] DMA Register Address Map ..... 505
Table 16-140 USB EHCI PCI Register Address Map (USB EHCI—D29:F7) ..... 519
Table 16-141 Enhanced Host Controller Capability Registers ..... 535
Table 16-142 Enhanced Host Controller Operational Register Address Map ..... 538
Table 16-143 Debug Port Register Address Map ..... 551
Table 17-144 SMBus Controller PCI Register Address Map (SMBUS—D31:F3) ..... 555
Table 17-145 SMBus I/O Register Address Map ..... 560
Table 18-146 Intel HD Audio PCI Register Address Map (Intel HD Audio D27:F0) ..... 573
Table 18-147 Intel HD Audio PCI Register Address Map (Intel HD Audio D27:F0) ..... 592
Table 19-148 PCI Express* Configuration Registers Address Map (PCI Express—D28:F0/F1/F2/F3) ..... 621
Table 20-149 Memory-Mapped Registers ..... 658
Table 21-150 Serial Peripheral Interface (SPI) Register Address Map (SPI Memory Mapped Configuration Registers) ..... 664



# Revision History

| Revision | Description | Date |
|---|---|---|
| 001 | • Initial release | December 2009 |

§

# 1 Introduction

This document provides specifications for the Intel® NM10 Family Express Chipset, which is designed for use in Intel's next generation Nettop platform.

***Note:*** In this document the Intel NM10 Family Express Chipset is referred to as the chipset.

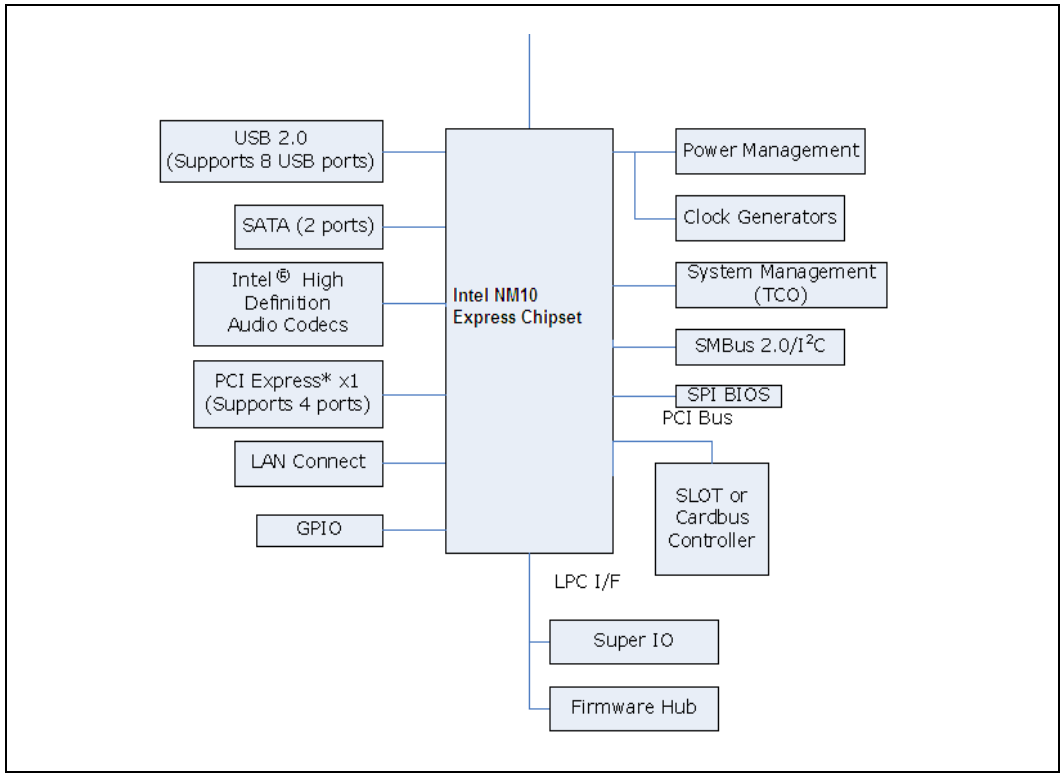
This document is intended for original equipment manufacturers and BIOS vendors. This manual assumes a working knowledge of the vocabulary and principles of PCI Express*, USB, AHCI, SATA, Intel® High Definition Audio (Intel HD Audio), SMBus, PCI, ACPI and LPC. Although some details of these features are described within this manual, refer to the individual industry specifications listed in Table 1-1 for the complete details.

**Table 1-1. Industry Specifications**

| Specification | Location |
|---|---|
| *PCI Express* Base Specification, Revision 1.0a* | http://www.pcisig.com/specifications |
| *Low Pin Count Interface Specification, Revision 1.1 (LPC)* | http://developer.intel.com/design/chipsets/industry/lpc.htm |
| *System Management Bus Specification, Version 2.0 (SMBus)* | http://www.smbus.org/specs/ |
| *PCI Local Bus Specification, Revision 2.3 (PCI)* | http://www.pcisig.com/specifications |
| *PCI Mobile Design Guide, Revision 1.1* | http://www.pcisig.com/specifications |
| *PCI Power Management Specification, Revision 1.1* | http://www.pcisig.com/specifications |
| *Universal Serial Bus Specification (USB), Revision 2.0* | http://www.usb.org/developers/docs |
| *Advanced Configuration and Power Interface, Version 2.0 (ACPI)* | http://www.acpi.info/spec.htm |
| *Universal Host Controller Interface, Revision 1.1 (UHCI)* | http://developer.intel.com/design/USB/UHCI11D.htm |
| *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0 (EHCI)* | http://developer.intel.com/technology/usb/ehcispec.htm |
| *Serial ATA Specification, Revision 1.0a* | http://www.serialata.org/specifications.asp |
| *Serial ATA II: Extensions to Serial ATA 1.0, Revision 1.0* | http://www.serialata.org/specifications.asp |
| *Alert Standard Format Specification, Version 1.03* | http://www.dmtf.org/standards/asf |
| *IEEE 802.3 Fast Ethernet* | http://standards.ieee.org/getieee802/ |
| *AT Attachment - 6 with Packet Interface (ATA/ATAPI - 6)* | http://T13.org (T13 1410D) |
| *IA-PC HPET (High Precision Event Timers) Specification, Revision 1.0* | http://www.intel.com/hardwaredesign/hpetspec.htm |

Figure 1-1 provides a block diagram of the Intel NM10 Series Express Chipset Family in the Nettop platform.

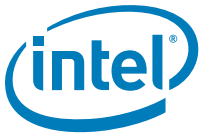
**Figure 1-1. Intel NM10 Family Express Chipset Features Block Diagram**




## 1.1 Intel® NM10 Family Express Chipset Feature Support

- Direct Media Interface
  - — 10 Gb/s each direction, full duplex
  - — Transparent to software
- PCI Express*
  - — 4 PCI Express root ports
  - — Supports PCI Express 1.0a
  - — Ports 1-4 can be statically configured as 4x1, or 1x4.
  - — Support for full 2.5 Gb/s bandwidth in each direction per x1 lane
  - — Module based Hot-Plug supported
- PCI Bus Interface
  - — Supports PCI Rev 2.3 Specification at 33 MHz

  - 2 available PCI REQ/GNT pairs Support for 64-bit addressing on PCI using DAC protocol
- Integrated Serial ATA Host Controller
  - Two ports
  - Data transfer rates up to 3.0 Gb/s (300 MB/s)
  - Integrated AHCI controller
- Intel HD Audio Interface
  - PCI Express endpoint
  - Independent Bus Master logic for eight general purpose streams: four input and four output
  - Support three external Codecs
  - Supports variable length stream slots
  - Supports multichannel, 32-bit sample depth and 192 kHz sample rate output
  - Provides mic array support
  - Allows for non-48 kHz sampling output
  - Support for ACPI Device States
- SMBus
  - Flexible SMBus/SMLink architecture to optimize for ASF
  - Provides independent manageability bus through SMLink interface
  - Supports SMBus 2.0 Specification
  - Host interface allows processor to communicate via SMBus
  - Slave interface allows an internal or external Microcontroller to access system resources
  - Compatible with most two-wire components that are also $I^2C$ compatible
- High Precision Event Timers
  - Advanced operating system interrupt scheduling
- Timers Based on 82C54
  - System timer, Refresh request, Speaker tone output
- Real-Time Clock
  - 256-byte battery-backed CMOS RAM
  - Integrated oscillator components
  - Lower Power DC/DC Converter implementation
- System TCO Reduction Circuits
  - Timers to generate SMI# and Reset upon detection of system hang
  - Timers to detect improper processor reset



  - — Integrated processor frequency strap logic
  - — Supports ability to disable external devices
- USB 2.0
  - — Includes four UHCI Host Controllers, supporting eight external ports
  - — Includes one EHCI Host Controller that supports all eight ports
  - — Includes one USB 2.0 High-speed Debug Port
  - — Supports wake-up from sleeping states S1–S5
  - — Supports legacy Keyboard/Mouse software
- Integrated LAN Controller
- Integrated ASF Management Controller
  - — Supports IEEE 802.3
  - — LAN Connect Interface (LCI)
  - — 10/100 Mb/s Ethernet Support
- Power Management Logic
  - — Supports ACPI 3.0
  - — ACPI-defined power states (C1, S1, S3–S5 for Nettop and C1–C4, S1, S3–S5 for Netbook)
  - — ACPI Power Management Timer
  - — Support for “Intel SpeedStep® Technology” processor power control and “Deeper Sleep” power state
  - — PCI CLKRUN# and PME# support
  - — SMI# generation
  - — All registers readable/restorable for proper resume from 0 V suspend states
  - — Support for APM-based legacy power management for non-ACPI Nettop and Netbook implementations
- External Glue Integration
  - — Integrated Pull-up, Pull-down and Series Termination resistors on processor interface
  - — Integrated Pull-down and Series resistors on USB
- Enhanced DMA Controller
  - — Two cascaded 8237 DMA controllers
  - — Supports LPC DMA
- Interrupt Controller
  - — Supports up to eight PCI interrupt pins
  - — Supports PCI 2.3 Message Signaled Interrupts
  - — Two cascaded 82C59 with 15 interrupts

  - — Integrated I/O APIC capability with 24 interrupts
  - — Supports Processor System Bus interrupt delivery
- 1.05 V operation with 1.5 V and 3.3 V I/O
  - — 5 V tolerant buffers on PCI, USB and Legacy signals
- 1.05 V Core Voltage
- Integrated 1.05 V Voltage Regulator (INTVR) for the Suspend and LAN wells Firmware Hub I/F supports BIOS Memory size up to 8 MBytes
- Serial Peripheral Interface (SPI) for Serial and Shared Flash
- Low Pin Count (LPC) I/F
  - — Supports two Master/DMA devices.
  - — Support for Security Device (Trusted Platform Module) connected to LPC.
- GPIO
  - — TTL, Open-Drain, Inversion
- Package 17 mm x17 mm 360 MMAP

## 1.2 Content Layout

**Chapter 1. Introduction**

This chapter introduces Intel NM10 Express Chipset and provides information on manual organization and gives a general overview of chipset.

**Chapter 2. Signal Description**

This chapter provides a block diagram of Intel NM10 Express Chipset interface signals and a detailed description of each signal. Signals are arranged according to interface and details are provided as to the drive characteristics (Input/Output, Open Drain, etc.) of all signals.

**Chapter 3. Intel NM10 Express Chipset Pin States**

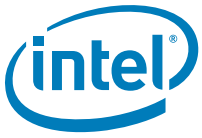
This chapter provides a complete list of signals, their associated power well, their logic level in each suspend state, and their logic level before and after reset.

**Chapter 4. Intel NM10 Express Chipset and System Clock Domains**

This chapter provides a list of each clock domain associated with Intel NM10 Express chipset in a chipset-based system.

**Chapter 5. Functional Description**

This chapter provides a detailed description of the functions in the Intel NM10 Express Chipset. All PCI buses, devices and functions in this manual are abbreviated using the following nomenclature; Bus:Device:Function. This manual abbreviates buses as B0 and B1, devices as D8, D27, D28, D29, D30 and D31 and functions as F0, F1, F2, F3, F4, F5, F6 and F7. 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. Note that Intel NM10 Express Chipset's external PCI bus is typically Bus 1, but may be assigned a different number depending upon system configuration.

**Chapter 6. Ballout Definition**

This chapter provides a table of each signal and its ball assignment in the 360-MMAP package.

**Chapter 7. Package Information**

This chapter provides drawings of the physical dimensions and characteristics of the 360-MMAP package.

**Chapter 8. Electrical Characteristics**

This chapter provides all AC and DC characteristics including detailed timing diagrams.

**Chapter 9. Register and Memory Mappings**

This chapter provides an overview of the registers, fixed I/O ranges, variable I/O ranges and memory ranges decoded by Intel NM10 Express Chipset.

**Chapter 10. Chipset Configuration Registers**

This chapter provides a detailed description of all registers and base functionality that is related to chipset configuration and not a specific interface (such as LPC, PCI, or PCI Express). It contains the root complex register block, which describes the behavior of the upstream internal link.

**Chapter 11. LAN Controller Registers**

This chapter provides a detailed description of all registers that reside in Intel NM10 Express Chipset's integrated LAN controller. The integrated LAN controller resides on Chipset's external PCI bus (typically Bus 1) at Device 8, Function 0 (B1:D8:F0).

**Chapter 12. PCI-to-PCI Bridge Registers**

This chapter provides a detailed description of all registers that reside in the PCI-to-PCI bridge. This bridge resides at Device 30, Function 0 (D30:F0).

**Chapter 13. LPC Bridge Registers**

This chapter provides a detailed description of all registers that reside in the LPC bridge. This bridge resides at Device 31, Function 0 (D31:F0). This function contains registers for many different units within Intel NM10 Express Chipset including DMA, Timers, Interrupts, Processor Interface, GPIO, Power Management, System Management and RTC.

**Chapter 14. SATA Controller Registers**

This chapter provides a detailed description of all registers that reside in the SATA controller. This controller resides at Device 31, Function 2 (D31:F2).

**Chapter 15. UHCI Controller Registers**

This chapter provides a detailed description of all registers that reside in the four UHCI host controllers. These controllers reside at Device 29, Functions 0, 1, 2, and 3 (D29:F0/F1/F2/F3).

**Chapter 16. EHCI Controller Registers**

This chapter provides a detailed description of all registers that reside in the EHCI host controller. This controller resides at Device 29, Function 7 (D29:F7).

**Chapter 17. SMBus Controller Registers**

This chapter provides a detailed description of all registers that reside in the SMBus controller. This controller resides at Device 31, Function 3 (D31:F3).

**Chapter 18.** Intel HD Audio **Controller Registers**

This chapter provides a detailed description of all registers that reside in the Intel HD Audio controller. This controller resides at Device 27, Function 0 (D27:F0).

**Chapter 19. PCI Express* Port Controller Registers**

This chapter provides a detailed description of all registers that reside in the PCI Express controller. This controller resides at Device 28, Functions 0 to 3(D30:F0-F3).

**Chapter 20. High** Precision E**vent Timers Registers**

This chapter provides a detailed description of all registers that reside in the multimedia timer memory mapped register space.

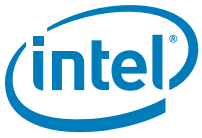
**Chapter 21. Serial Peripheral Interface Registers**

This chapter provides a detailed description of all registers that reside in the SPI memory mapped register space.

## 1.3 Functions and capabilities

This chipset provides extensive I/O support. Functions and capabilities include:

- *PCI Express* Base Specification,* Revision 1.0a support
- *PCI Local Bus Specification*, Revision 2.3 support for 33 MHz PCI operations
- ACPI Power Management Logic Support
- Enhanced DMA controller, interrupt controller, and timer functions
- Integrated Serial ATA host controller with independent DMA operation on two ports and AHCI support.
- USB host interface with support for eight USB ports; four UHCI host controllers; one EHCI high-speed USB 2.0 Host controller
- Integrated LAN controller
- *System Management Bus (SMBus) Specification*, Version 2.0 with additional support for $I^2C$ devices
- Supports Intel High Definition Audio
- Supports Intel® Matrix Storage Technology
- Low Pin Count (LPC) interface



- Firmware Hub (FWH) interface support
- Serial Peripheral Interface (SPI) support

This chipset incorporates a variety of PCI functions that are divided into six logical devices (B0:D27, B0:D28, B0:D29, B0:D30, B0:D31 and B1:D8) as listed in Table 1-2. D30 is the DMI-to-PCI bridge, D31 contains the PCI-to-LPC bridge, SATA controller, and SMBus controller, D29 contains the four USB UHCI controllers and one USB EHCI controller, and D27 contains the PCI Express root ports. B1:D8 is the integrated LAN controller.

**Table 1-2. PCI Devices and Functions**

| Bus:Device:Function | Function Description |
|---|---|
| Bus 0:Device 30:Function 0 | PCI-to-PCI Bridge |
| Bus 0:Device 31:Function 0 | LPC Controller[1] |
| Bus 0:Device 31:Function 2 | SATA Controller |
| Bus 0:Device 31:Function 3 | SMBus Controller |
| Bus 0:Device 29:Function 0 | USB UHCI Controller #1 |
| Bus 0:Device 29:Function 1 | USB UHCI Controller #2 |
| Bus 0:Device 29:Function 2 | USB UHCI Controller #3 |
| Bus 0:Device 29:Function 3 | USB UHCI Controller #4 |
| Bus 0:Device 29:Function 7 | USB 2.0 EHCI Controller |
| Bus 0:Device 28:Function 0 | PCI Express* Port 1 |
| Bus 0:Device 28:Function 1 | PCI Express Port 2 |
| Bus 0:Device 28:Function 2 | PCI Express Port 3 |
| Bus 0:Device 28:Function 3 | PCI Express Port 4 |
| Bus 0:Device 27:Function 0 | Intel HD Audio Controller |
| Bus n:Device 8:Function 0 | LAN Controller |

**NOTES:**
1 The PCI-to-LPC bridge contains registers that control LPC, Power Management, System Management, GPIO, Processor Interface, RTC, Interrupts, Timers, and DMA.

The following sub-sections provide an overview of Intel NM10 Express Chipset capabilities.

### Direct Media Interface (DMI)

Direct Media Interface (DMI) is the chip-to-chip connection between the Processor or MCH and chipset. This high-speed interface integrates advanced priority-based servicing allowing for concurrent traffic and true isochronous transfer capabilities. Base functionality is completely software-transparent, permitting current and legacy software to operate normally.

### PCI Express* Interface

Intel NM10 Express Chipset has four PCI Express root ports (ports 1-4), supporting the *PCI Express Base Specification,* Revision 1.0a. PCI Express root ports 1–4 can be statically configured as four x1 ports or ganged together to form one x4 port. Each Root Port supports 2.5 Gb/s bandwidth in each direction (5 Gb/s concurrent).

### Serial ATA (SATA) Controller

Intel NM10 Express Chipset has an integrated SATA host controller that supports independent DMA operation on two ports and supports data transfer rates of up to 3.0 Gb/s (300 MB/s). The SATA controller contains two modes of operation – a legacy mode using I/O space, and an AHCI mode using memory space.

Intel NM10 Express Chipset supports the *Serial ATA Specification*, Revision 1.0a.This chipset also supports several optional sections of the Serial ATA II: Extensions to *Serial ATA 1.0 Specification*, Revision 1.0 (AHCI support is required for some elements).

### AHCI

Intel NM10 Express Chipset provides hardware support for Advanced Host Controller Interface (AHCI), a new programming interface for SATA host controllers. Platforms supporting AHCI may take advantage of performance features such as no master/slave designation for SATA devices—each device is treated as a master—and hardware-assisted native command queuing. AHCI also provides usability enhancements such as Hot-Plug. AHCI requires appropriate software support (e.g., an AHCI driver) and for some features, hardware support in the SATA device or additional platform hardware.

### PCI Interface

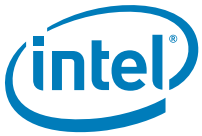
Intel NM10 Express Chipset PCI interface provides a 33 MHz, Revision 2.3 implementation. This chipset integrates a PCI arbiter that supports up to two external PCI bus masters in addition to the internal chipset requests. This allows for combinations of up to two PCI down devices and PCI slots.

### Low Pin Count (LPC) Interface

Intel NM10 Express Chipset implements an LPC Interface as described in the *LPC 1.1 Specification*. The Low Pin Count (LPC) bridge function of this chipset resides in PCI Device 31:Function 0. In addition to the LPC bridge interface function, D31:F0 contains other functional units including DMA, interrupt controllers, timers, power management, system management, GPIO, and RTC.

### Serial Peripheral Interface (SPI)

Intel NM10 Express Chipset implements an SPI Interface as an alternative interface for the BIOS flash device. An SPI flash device can be used as a replacement for the FWH.



### Compatibility Modules (DMA Controller, Timer/Counters, Interrupt Controller)

The DMA controller incorporates the logic of two 82C37 DMA controllers, with seven independently programmable channels. Channels 0–3 are hardwired to 8-bit, count-by-byte transfers, and channels 5–7 are hardwired to 16-bit, count-by-word transfers. Any two of the seven DMA channels can be programmed to support fast Type-F transfers.

Intel NM10 Express Chipset supports LPC DMA, which is similar to ISA DMA, through Chipset's DMA controller. LPC DMA is handled through the use of the LDRQ# lines from peripherals and special encoding on LAD[3:0] from the host. Single, Demand, Verify, and Increment modes are supported on the LPC interface. Channels 0–3 are 8-bit channels. Channels 5–7 are 16-bit channels. Channel 4 is reserved as a generic bus master request.

The timer/counter block contains three counters that are equivalent in function to those found in one 82C54 programmable interval timer. These three counters are combined to provide the system timer function, and speaker tone. The 14.31818 MHz oscillator input provides the clock source for these three counters.

Intel NM10 Express Chipset provides an ISA-Compatible Programmable Interrupt Controller (PIC) that incorporates the functionality of two, 82C59 interrupt controllers. The two interrupt controllers are cascaded so that 14 external and two internal interrupts are possible. In addition, this chipset supports a serial interrupt scheme.

All of the registers in these modules can be read and restored. This is required to save and restore system state after power has been removed and restored to the platform.

### Advanced Programmable Interrupt Controller (APIC)

In addition to the standard ISA compatible Programmable Interrupt controller (PIC) described in the previous section, this chipset incorporates the Advanced Programmable Interrupt Controller (APIC).

### Universal Serial Bus (USB) Controller

Intel NM10 Express Chipset contains an Enhanced Host Controller Interface (EHCI) host controller that supports USB high-speed signaling. High-speed USB 2.0 allows data transfers up to 480 Mb/s which is 40 times faster than full-speed USB. This chipset also contains four Universal Host Controller Interface (UHCI) controllers that support USB full-speed and low-speed signaling.

Intel NM10 Express Chipset supports eight USB 2.0 ports. All eight ports are high-speed, full-speed, and low-speed capable. This chipset's port-routing logic determines whether a USB port is controlled by one of the UHCI controllers or by the EHCI controller. See Section 5.18 and Section 5.19 for details.

### LAN Controller

Intel NM10 Express Chipset's integrated LAN controller includes a 32-bit PCI controller that provides enhanced scatter-gather bus mastering capabilities and enables the LAN controller to perform high speed data transfers over the PCI bus. Its bus master capabilities enable the component to process high-level commands and perform multiple operations; this lowers processor utilization by off-loading communication tasks from the processor. Two large transmit and receive FIFOs of 3 KB each help prevent data underruns and overruns while waiting for bus accesses. This enables the integrated LAN controller to transmit data with minimum interframe spacing (IFS).

The LAN controller can operate in either full duplex or half duplex mode. In full duplex mode the LAN controller adheres with the *IEEE 802.3x Flow Control* Specification. Half duplex performance is enhanced by a proprietary collision reduction mechanism. See Section 5.3 for details.

### Alert Standard Format (ASF) Management Controller

Intel NM10 Express Chipset integrates an Alert Standard Format controller in addition to the integrated LAN controller, allowing interface system-monitoring devices to communicate through the integrated LAN controller to the network. This makes remote manageability and system hardware monitoring possible using ASF.

The ASF controller can collect and send various information from system components such as the processor, chipset, BIOS and sensors on the motherboard to a remote server running a management console. The controller can also be programmed to accept commands back from the management console and execute those commands on the local system.

### RTC

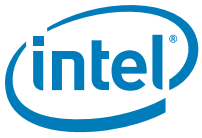
Intel NM10 Express Chipset contains a Motorola* MC146818A-compatible real-time clock with 256 bytes of battery-backed RAM. The real-time clock performs two key functions: keeping track of the time of day and storing system data, even when the system is powered down. The RTC operates on a 32.768 KHz crystal and a 3 V battery.

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

The RTC also supports a date alarm that allows for scheduling a wake up event up to 30 days in advance, rather than just 24 hours in advance.

### GPIO

Various general purpose inputs and outputs are provided for custom system design. The number of inputs and outputs varies depending on Intel NM10 Express Chipset configuration.



### Enhanced Power Management

Intel NM10 Express Chipset's power management functions include enhanced clock control and various low-power (suspend) states (e.g., Suspend-to-RAM and Suspend-to-Disk). A hardware-based thermal management circuit permits software-independent entrance to low-power states. This chipset contains full support for the *Advanced Configuration and Power Interface (ACPI) Specification,* Revision 3.0.

Manageability

Intel NM10 Express Chipset integrates several functions designed to manage the system and lower the total cost of ownership (TCO) of the system. These system management functions are designed to report errors, diagnose the system, and recover from system lockups without the aid of an external microcontroller.

- **TCO Timer.** Chipset's integrated programmable TCO timer is used to detect system locks. The first expiration of the timer generates an SMI# that the system can use to recover from a software lock. The second expiration of the timer causes a system reset to recover from a hardware lock.
- **Processor Present Indicator.** Chipset looks for the processor to fetch the first instruction after reset. If the processor does not fetch the first instruction, Chipset will reboot the system.
- **ECC Error Reporting.** When detecting an ECC error, the host controller has the ability to send one of several messages to Chipset. The host controller can instruct Chipset to generate either an SMI#, NMI, SERR#, or TCO interrupt.
- **Function Disable.** Chipset provides the ability to disable the following integrated functions: LAN, USB, LPC, Intel HD Audio, SATA, or SMBus. Once disabled, these functions no longer decode I/O, memory, or PCI configuration space. Also, no interrupts or power management events are generated from the disable functions.
- **Intruder Detect.** Chipset provides an input signal (INTRUDER#) that can be attached to a switch that is activated by the system case being opened. Chipset can be programmed to generate an SMI# or TCO interrupt due to an active INTRUDER# signal.

### System Management Bus (SMBus 2.0)

This chipset contains an SMBus Host interface that allows the processor to communicate with SMBus slaves. This interface is compatible with most $I^2C$ devices. Special $I^2C$ commands are implemented.

This chipset's SMBus host controller provides a mechanism for the processor to initiate communications with SMBus peripherals (slaves). Also, this chipset supports slave functionality, including the Host Notify protocol. Hence, the host controller supports eight command protocols of the SMBus interface (see *System Management Bus (SMBus) Specification,* Version 2.0): Quick Command, Send Byte, Receive Byte, Write Byte/Word, Read Byte/Word, Process Call, Block Read/Write, and Host Notify.

Chipset's SMBus also implements hardware-based Packet Error Checking for data robustness and the Address Resolution Protocol (ARP) to dynamically provide address to all SMBus devices.

### Intel® HD Audio Controller (Intel® HD Audio)

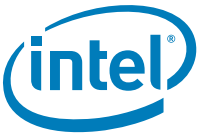
The *Intel HD Audio Specification* defines a digital interface that can be used to attach different types of codecs, such as audio and modem codecs. In this chipset Intel HD Audio digital link shares pins with the AC-link. Chipset's Intel HD Audio controller supports up to three codecs.

With the support of multi-channel audio stream, 32-bit sample depth, and sample rate up to 192 kHz, the Intel HD Audio controller provides audio quality that can deliver CE levels of audio experience. On the input side, chipset adds support for an arrays of microphones.

The Intel HD Audio controller uses multi-purpose DMA engines, to effectively manage the link bandwidth and support simultaneous independent streams on the link. The capability enables new exciting usage models with Intel HD Audio (e.g., listening to music while playing multi-player game on the internet.) The Intel HD Audio controller also supports isochronous data transfers allowing glitch-free audio to the system.

***Note:*** Users interested in providing feedback on the *Intel High Definition Audio Specification* or planning to implement the *Intel High Definition Audio Specification* into a future product will need to execute the *Intel High Definition Audio Specification Developer's Agreement*. For more information, contact nextgenaudio@intel.com.

§



# 2 Signal Description

This chapter provides a detailed description of each signal. The signals are arranged in functional groups according to their associated interface. Figure 2-2 and Section 2.1 shows the interface signals for chipset.

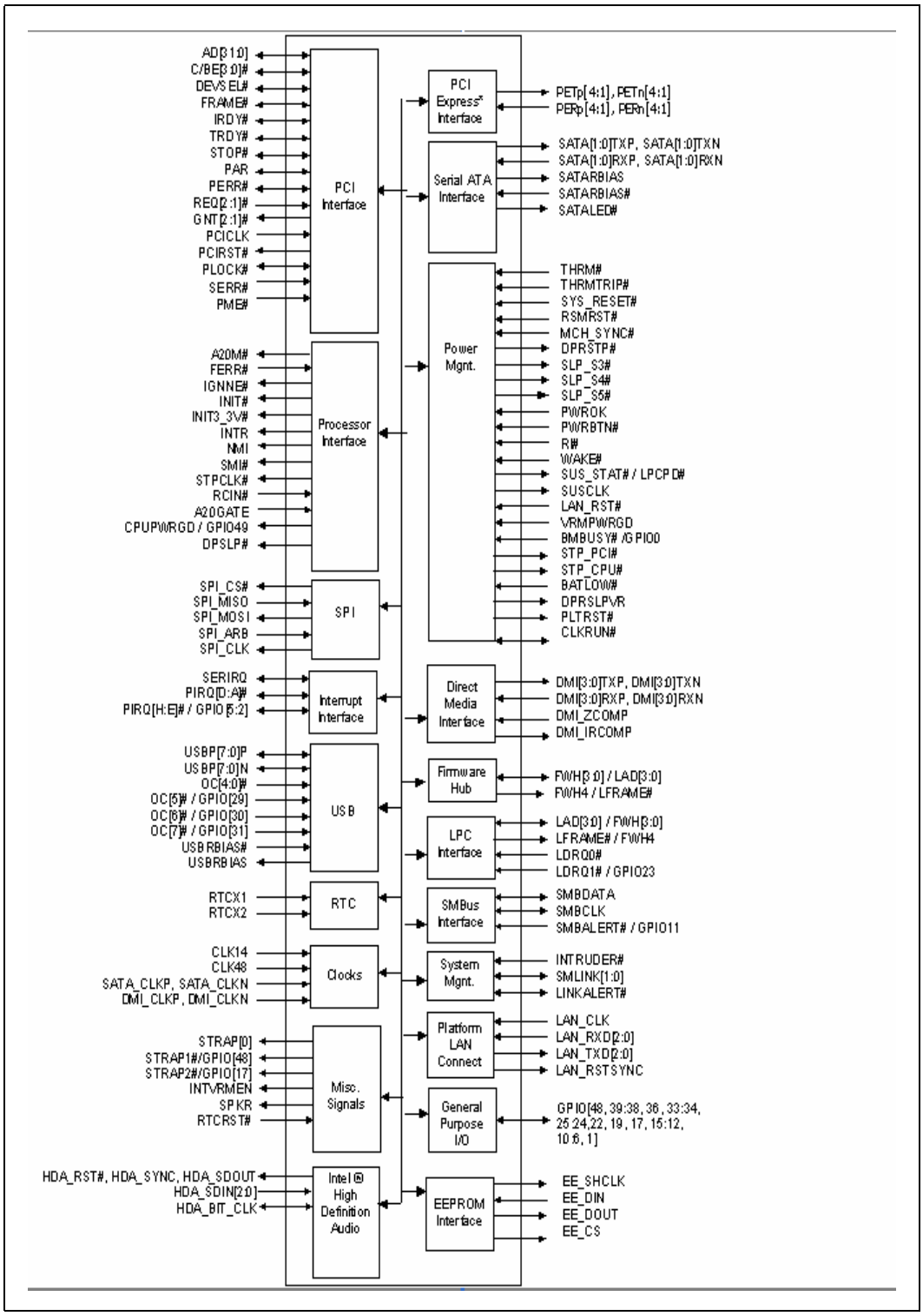
The "#" symbol at the end of the signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When "#" is not present, the signal is asserted when at the high voltage level.

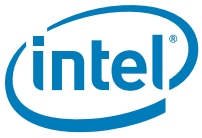
The "Type" for each signal is indicative of the functional operating mode of the signal. Unless otherwise noted in Section 3.2 or Section 3.3, a signal is considered to be in the functional operating mode after RTCRST# for signals in the RTC well, RSMRST# for signals in the suspend well, after PWROK for signals in the core well, and after LAN_RST# for signals in the LAN well.

The following notations are used to describe the signal type:

| | |
|---|---|
| **I** | Input Pin |
| **O** | Output Pin |
| **OD O** | Open Drain Output Pin. |
| **I / OD** | Bi-directional Input/Open Drain Output Pin. |
| **I / O** | Bi-directional Input / Output Pin. |
| **OC** | Open Collector Output Pin. |

## Figure 2-2. Interface Signals Block Diagram





## 2.1 Direct Media Interface (DMI) to Host Controller

**Table 2-3. Direct Media Interface Signals**

| Name | Type | Description |
|---|---|---|
| DMI[0:1]TXP, DMI[0:1]TXN DMI[2:3]TXP, DMI[2:3]TXN | O | Direct Media Interface Differential Transmit Pair 0:3 |
| DMI[0:1]RXP, DMI[0:1]RXN DMI[2:3]RXP, DMI[2:3]RXN | I | Direct Media Interface Differential Receive Pair 0:3 |
| DMI_ZCOMP | I | **Impedance Compensation Input:** Determines DMI input impedance. |
| DMI_IRCOMP | O | **Impedance/Current Compensation Output:** Determines DMI output impedance and bias current. |

## 2.2 PCI Express*

**Table 2-4. PCI Express* Signals**

| Name | Type | Description |
|---|---|---|
| PETp[1:4], PETn[1:4] | O | PCI Express* Differential Transmit Pair 1:4 |
| PERp[1:4], PERn[1:4] | I | PCI Express Differential Receive Pair 1:4 |

## 2.3 Platform LAN Connect Interface

**Table 2-5. Platform LAN Connect Interface Signals**

| Name | Type | Description |
|---|---|---|
| LAN_CLK | I | **LAN I/F Clock:** This signal is driven by the Platform LAN Connect component. The frequency range is 5 MHz to 50 MHz. |
| LAN_RXD[2:0] | I | **Received Data:** The Platform LAN Connect component uses these signals to transfer data and control information to the integrated LAN controller. These signals have integrated weak pull-up resistors. |
| LAN_TXD[2:0] | O | **Transmit Data**: The integrated LAN controller uses these signals to transfer data and control information to the Platform LAN Connect component. |
| LAN_RSTSYNC | O | **LAN Reset/Sync:** The Platform LAN Connect component's Reset and Sync signals are multiplexed onto this pin. |

## 2.4 EEPROM Interface

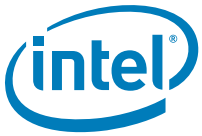
**Table 2-6. EEPROM Interface Signals**

| Name | Type | Description |
|---|---|---|
| EE_SHCLK | O | **EEPROM Shift Clock**: This signal is the serial shift clock output to the EEPROM. |
| EE_DIN | I | **EEPROM Data In**: This signal transfers data from the EEPROM to chipset. This signal has an integrated pull-up resistor. |
| EE_DOUT | O | **EEPROM Data Out**: This signal transfers data from chipset to the EEPROM. |
| EE_CS | O | **EEPROM Chip Select**: This is the chip select signal to the EEPROM. |

## 2.5 Firmware Hub Interface

**Table 2-7. Firmware Hub Interface Signals**

| Name | Type | Description |
|---|---|---|
| FWH[3:0] / LAD[3:0] | I/O | Firmware Hub Signals: These signals are multiplexed with the LPC address signals. |
| FWH4 / LFRAME# | O | Firmware Hub Signals: This signal is multiplexed with the LPC LFRAME# signal. |



## 2.6 PCI Interface

**Table 2-8. PCI Interface Signals (Sheet 1 of 3)**

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>AD[31:0]</td><td>I/O</td><td>PCI Address/Data: AD[31:0] is a multiplexed address and data bus. During the first clock of a transaction, AD[31:0] contain a physical address (32 bits). During subsequent clocks, AD[31:0] contain data. chipset will drive all 0s on AD[31:0] during the address phase of all PCI Special Cycles.</td></tr>
<tr><td>C/BE[3:0]#</td><td>I/O</td><td>Bus Command and Byte Enables: The command and byte enable signals are multiplexed on the same PCI pins. During the address phase of a transaction, C/BE[3:0]B define the bus command. During the data phase, C/BE[3:0]B define the Byte Enables.<br>C/BE[3:0]# Command Type<br>0000b Interrupt Acknowledge<br>0001b Special Cycle<br>0010b I/O Read<br>0011b I/O Write<br>0110b Memory Read<br>0111b Memory Write<br>1010b Configuration Read<br>1011b Configuration Write<br>1100b Memory Read Multiple<br>1110b Memory Read Line<br>1111b Memory Write and Invalidate<br>All command encodings not shown are reserved. chipset does not decode reserved values, and therefore will not respond if a PCI master generates a cycle using one of the reserved values.</td></tr>
<tr><td>DEVSEL#</td><td>I/O</td><td>Device Select: chipset asserts DEVSELB to claim a PCI transaction. As an output, chipset asserts DEVSEL# when a PCI master peripheral attempts an access to an internal chipset address or an address destined DMI (main memory or graphics). As an input, DEVSEL# indicates the response to an chipset-initiated transaction on the PCI bus. DEVSEL# is tri-stated from the leading edge of PLTRST#. DEVSEL# remains tri-stated by chipset until driven by a target device.</td></tr>
<tr><td>FRAME#</td><td>I/O</td><td>Cycle Frame: The current initiator drives FRAME# to indicate the beginning and duration of a PCI transaction. While the initiator asserts FRAME#, data transfers continue. When the initiator negates FRAME#, the transaction is in the final data phase. FRAME# is an input to chipset when chipset is the target, and FRAME# is an output from chipset when chipset is the initiator. FRAME# remains tri-stated by chipset until driven by an initiator.</td></tr>
<tr><td>IRDY#</td><td>I/O</td><td>Initiator Ready: IRDY# indicates chipset's ability, as an initiator, to complete the current data phase of the transaction. It is used in conjunction with TRDY#. A data phase is completed on any clock both IRDY# and TRDY# are sampled asserted. During a write, IRDY# indicates chipset has valid data present on AD[31:0]. During a read, it indicates chipset is prepared to latch data. IRDY# is an input to chipset when chipset is the target and an output from chipset when chipset is an initiator. IRDY# remains tri-stated by chipset until driven by an initiator.</td></tr>
</table>

**Table 2-8. PCI Interface Signals (Sheet 2 of 3)**

| Name | Type | Description |
|---|---|---|
| TRDY# | I/O | **Target Ready**: TRDY# indicates chipset's ability as a target to complete the current data phase of the transaction. TRDY# is used in conjunction with IRDY#. A data phase is completed when both TRDY# and IRDY# are sampled asserted. During a read, TRDYB indicates that chipset, as a target, has placed valid data on AD[31:0]. During a write, TRDY# indicates chipset, as a target is prepared to latch data. TRDY# is an input to chipset when chipset is the initiator and an output from chipset when chipset is a target. TRDY# is tri-stated from the leading edge of PLTRST#. TRDY# remains tri-stated by chipset until driven by a target. |
| STOP# | I/O | **Stop**: STOP# indicates that chipset, as a target, is requesting the initiator to stop the current transaction. STOP# causes chipset, as an initiator, to stop the current transaction. STOPB is an output when chipset is a target and an input when chipset is an initiator. |
| PAR | I/O | **Calculated/Checked Parity:** PAR uses “even” parity calculated on 36 bits, AD[31:0] plus C/BE[3:0]B. “Even” parity means that chipset counts the number of 1s within the 36 bits plus PAR and the sum is always even. chipset calculates PAR on 36 bits regardless of the valid byte enables. chipset generates PAR for address and data phases and only ensures PAR to be valid one PCI clock after the corresponding address or data phase. chipset drives and tri-states PAR identically to the AD[31:0] lines except that chipset delays PAR by exactly one PCI clock. PAR is an output during the address phase (delayed one clock) for all chipset initiated transactions. PAR is an output during the data phase (delayed one clock) when chipset is the initiator of a PCI write transaction, and when it is the target of a read transaction. chipset checks parity when it is the target of a PCI write transaction. If a parity error is detected, chipset will set the appropriate internal status bits, and has the option to generate an NMIB or SMIB. |
| PERR# | I/O | **Parity Error**: An external PCI device drives PERRB when it receives data that has a parity error. chipset drives PERRB when it detects a parity error. chipset can either generate an NMIB or SMIB upon detecting a parity error (either detected internally or reported via the PERR# signal). |
| REQ[2:1] | I | **PCI Requests:** chipset supports up to 2 masters on the PCI bus. |
| GNT[2:1]# | O | **PCI Grants:** chipset supports up to 2 masters on the PCI bus.<br>Pull-up resistors are not required on these signals. If pull-ups are used, they should be tied to the Vcc3_3 power rail. |
| PCICLK | I | **PCI Clock**: This is a 33 MHz clock. PCICLK provides timing for all transactions on the PCI Bus. |
| PCIRST# | O | **PCI Reset:** This is the Secondary PCI Bus reset signal. It is a logical OR of the primary interface PLTRST# signal and the state of the Secondary Bus Reset bit of the Bridge Control register (D30:F0:3Eh, bit 6). |

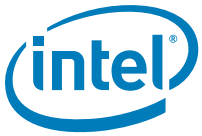


**Table 2-8. PCI Interface Signals (Sheet 3 of 3)**

| Name | Type | Description |
|---|---|---|
| PLOCK# | I/O | **PCI Lock**: This signal indicates an exclusive bus operation and may require multiple transactions to complete. chipset asserts PLOCK# when it performs non-exclusive transactions on the PCI bus. PLOCK# is ignored when PCI masters are granted the bus in Nettop configurations. Devices on the PCI bus (other than chipset) are not permitted to assert the PLOCK# signal in Netbook configurations. |
| SERR# | I/OD | **System Error**: SERR# can be pulsed active by any PCI device that detects a system error condition. Upon sampling SERR# active, chipset has the ability to generate an NMI, SMI#, or interrupt. |
| PME# | I/OD | **PCI Power Management Event**: PCI peripherals drive PME# to wake the system from low-power states S1–S5. PME# assertion can also be enabled to generate an SCI from the S0 state. In some cases chipset may drive PME# active due to an internal wake event. chipset will not drive PME# high, but it will be pulled up to VccSus3_3 by an internal pull-up resistor. |

## 2.7 Serial ATA Interface

**Table 2-9. Serial ATA Interface Signals**

| Name | Type | Description |
|---|---|---|
| SATA0TXP<br>SATA0TXN | O | **Serial ATA 0 Differential Transmit Pair:** These are outbound high-speed differential signals to Port 0. |
| SATA0RXP<br>SATA0RXN | I | **Serial ATA 0 Differential Receive Pair:** These are inbound high-speed differential signals from Port 0. |
| SATA1TXP<br>SATA1TXN | O | **Serial ATA 1 Differential Transmit Pair:** These are outbound high-speed differential signals to Port 1. |
| SATA1RXP<br>SATA1RXN | I | **Serial ATA 1 Differential Receive Pair:** These are inbound high-speed differential signals from Port 1. |
| SATARBIAS | O | **Serial ATA Resistor Bias:** These are analog connection points for an external resistor to ground. |
| SATARBIAS# | I | **Serial ATA Resistor Bias Complement:** These are analog connection points for an external resistor to ground. |
| SATALED# | OC | Serial ATA LED: This is an open-collector 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. An external pull-up resistor to Vcc3_3 is required.<br>**NOTE:** An internal pull-up is enabled only during PLTRST# assertion. |

## 2.8 LPC Interface

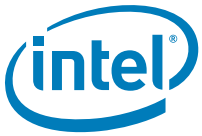
**Table 2-10.LPC Interface Signals**

| Name | Type | Description |
|---|---|---|
| LAD[3:0] / FWH[3:0] | I/O | **LPC Multiplexed Command, Address, Data:** For LAD[3:0], internal pull-ups are provided. |
| LFRAME#/ FWH4 | O | **LPC Frame:** LFRAME# indicates the start of an LPC cycle, or an abort. |
| LDRQ0# LDRQ1# / GPIO23 | I | **LPC Serial DMA/Master Request Inputs:** LDRQ[1:0]# are used to request DMA or bus master access. These signals are typically connected to external Super I/O device. An internal pull-up resistor is provided on these signals.<br>LDRQ1# may optionally be used as GPIO. |

## 2.9 Interrupt Interface

**Table 2-11.Interrupt Signals**

| Name | Type | Description |
|---|---|---|
| SERIRQ | I/O | **Serial Interrupt Request:** This pin implements the serial interrupt protocol. |
| PIRQ[D:A]# | I/OD | **PCI Interrupt Requests:** In non-APIC mode the PIRQx# signals can be routed to interrupts 3, 4, 5, 6, 7, 9, 10, 11, 12, 14 or 15 as described in the Interrupt Steering section. Each PIRQx# line has a separate Route Control register.<br>In APIC mode, these signals are connected to the internal I/O APIC in the following fashion: PIRQA# is connected to IRQ16, PIRQB# to IRQ17, PIRQC# to IRQ18, and PIRQD# to IRQ19. This frees the legacy interrupts. |
| PIRQ[H:E]# / GPIO[5:2] | I/OD | **PCI Interrupt Requests:** In non-APIC mode the PIRQx# signals can be routed to interrupts 3, 4, 5, 6, 7, 9, 10, 11, 12, 14 or 15 as described in the *Interrupt Steering* section. Each PIRQx# line has a separate Route Control register.<br>In APIC mode, these signals are connected to the internal I/O APIC in the following fashion: PIRQE# is connected to IRQ20, PIRQF# to IRQ21, PIRQG# to IRQ22, and PIRQH# to IRQ23. This frees the legacy interrupts. If not needed for interrupts, these signals can be used as GPIO. |



## 2.10 USB Interface

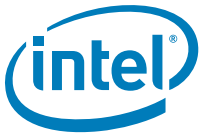
**Table 2-12.USB Interface Signals**

| Name | Type | Description |
|---|---|---|
| USBP0P,<br>USBP0N,<br>USBP1P,<br>USBP1N | I/O | **Universal Serial Bus Port [1:0] Differential**: These differential pairs are used to transmit Data/Address/Command signals for ports 0 and 1. These ports can be routed to UHCI controller #1 or the EHCI controller.<br>**NOTE:** No external resistors are required on these signals. The chipset integrates 15 kΩ pull-downs and provides an output driver impedance of 45 Ω which requires no external series resistor |
| USBP2P,<br>USBP2N,<br>USBP3P,<br>USBP3N | I/O | **Universal Serial Bus Port [3:2] Differential**: These differential pairs are used to transmit data/address/command signals for ports 2 and 3. These ports can be routed to UHCI controller #2 or the EHCI controller.<br>**NOTE:** No external resistors are required on these signals. chipset integrates 15 kΩ pull-downs and provides an output driver impedance of 45 Ω which requires no external series resistor |
| USBP4P,<br>USBP4N,<br>USBP5P,<br>USBP5N | I/O | **Universal Serial Bus Port [5:4] Differential**: These differential pairs are used to transmit Data/Address/Command signals for ports 4 and 5. These ports can be routed to UHCI controller #3 or the EHCI controller.<br>**NOTE:** No external resistors are required on these signals. chipset integrates 15 kΩ pull-downs and provides an output driver impedance of 45 Ω which requires no external series resistor |
| USBP6P,<br>USBP6N,<br>USBP7P,<br>USBP7N | I/O | **Universal Serial Bus Port [7:6] Differential**: These differential pairs are used to transmit Data/Address/Command signals for ports 6 and 7. These ports can be routed to UHCI controller #4 or the EHCI controller.<br>**NOTE:** No external resistors are required on these signals. The chipset integrates 15 kΩ pull-downs and provides an output driver impedance of 45 Ω which requires no external series resistor |
| OC[4:0]#<br>OC5# / GPIO29<br>OC6# / GPIO30<br>OC7# / GPIO31 | I | **Overcurrent Indicators:** These signals set corresponding bits in the USB controllers to indicate that an overcurrent condition has occurred.<br>OC[7:5]# may optionally be used as GPIOs.<br>**NOTE:** OC[7:0]# are not 5 V tolerant. |
| USBRBIAS | O | **USB Resistor Bias:** Analog connection point for an external resistor. Used to set transmit currents and internal load resistors. |
| USBRBIAS# | I | **USB Resistor Bias Complement:** Analog connection point for an external resistor. Used to set transmit currents and internal load resistors. |

## 2.11 Power Management Interface

**Table 2-13.Power Management Interface Signals (Sheet 1 of 3)**

| Name | Type | Description |
|---|---|---|
| PLTRST# | O | **Platform Reset:** The chipset asserts PLTRST# to reset devices on the platform (e.g., SIO, FWH, LAN, (G)MCH, IDE, TPM, etc.). The chipset asserts PLTRST# during power-up and when S/W initiates a hard reset sequence through the Reset Control register (I/O Register CF9h). The chipset drives PLTRST# inactive a minimum of 1 ms after both PWROK and VRMPWRGD are driven high. The chipset drives PLTRST# active a minimum of 1 ms when initiated through the Reset Control register (I/O Register CF9h).<br>**NOTE:** PLTRST# is in the VccSus3_3 well. |
| THRM# | I | **Thermal Alarm:** THRM# is an active low signal generated by external hardware to generate an SMI# or SCI. |
| THRMTRIP# | I | **Thermal Trip:** When low, this signal indicates that a thermal trip from the processor occurred, and the chipset will immediately transition to a S5 state. The chipset will not wait for the processor stop grant cycle since the processor has overheated. |
| SLP_S3# | O | **S3 Sleep Control:** SLP_S3# is for power plane control. This signal shuts off power to all non-critical systems when in S3 (Suspend To RAM), S4 (Suspend to Disk), or S5 (Soft Off) states. |
| SLP_S4# | O | S4 Sleep Control: SLP_S4# is for power plane control. This signal shuts power to all non-critical systems when in the S4 (Suspend to Disk) or S5 (Soft Off) state.<br>**NOTE:** This pin must be used to control the DRAM power to use chipset's DRAM power-cycling feature. Refer to Chapter 5.14.11.2 for details. |
| SLP_S5# | O | **S5 Sleep Control:** SLP_S5# is for power plane control. This signal is used to shut power off to all non-critical systems when in the S5 (Soft Off) states. |
| PWROK | I | **Power OK:** When asserted, PWROK is an indication to the chipset that core power has been stable for 99 ms and that PCICLK has been stable for 1 ms. An exception to this rule is if the system is in $S3_{HOT}$, in which PWROK may or may not stay asserted even though PCICLK may be inactive. PWROK can be driven asynchronously. When PWROK is negated, the chipset asserts PLTRST#.<br>**NOTE:** PWROK must deassert for a minimum of three RTC clock periods for the chipset to fully reset the power and properly generate the PLTRST# output. |
| PWRBTN# | I | **Power Button:** The Power Button will cause SMI# or SCI to indicate a system request to go to a sleep state. If the system is already in a sleep state, this signal will cause a wake event. If PWRBTN# is pressed for more than 4 seconds, this will cause an unconditional transition (power button override) to the S5 state. Override will occur even if the system is in the S1-S4 states. This signal has an internal pull-up resistor and has an internal 16 ms de-bounce on the input. |
| RI# | I | **Ring Indicate:** This signal is an input from a modem. It can be enabled as a wake event, and this is preserved across power failures. |



### Table 2-13.Power Management Interface Signals (Sheet 2 of 3)

| Name | Type | Description |
|---|---|---|
| SYS_RESET# | I | **System Reset:** This pin forces an internal reset after being debounced. The chipset will reset immediately if the SMBus is idle; otherwise, it will wait up to 25 ms ± 2 ms for the SMBus to idle before forcing a reset on the system. |
| RSMRST# | I | **Resume Well Reset:** This signal is used for resetting the resume power plane logic. |
| LAN_RST# | I | **LAN Reset:** When asserted, the internal LAN controller will be put into reset. This signal must be asserted for at least 10 ms after the resume well power is valid. When deasserted, this signal is an indication that the resume well power is stable.<br>**NOTE:** LAN_RST# should be tied to RSMRST#. |
| WAKE# | I | **PCI Express* Wake Event:** Sideband wake signal on PCI Express asserted by components requesting wake up. |
| MCH_SYNC# | I | **MCH SYNC:** This input is internally ANDed with the PWROK input. Connect to the ICH_SYNC# output of (G)MCH. This signal need to pull-up to VCC3_3 when the chipset is not pair with (G)MCH. |
| SUS_STAT# / LPCPD# | O | **Suspend Status:** This signal is asserted by The chipset to indicate that the system will be entering a low power state soon. This can be monitored by devices with memory that need to switch from normal refresh to suspend refresh mode. It can also be used by other peripherals as an indication that they should isolate their outputs that may be going to powered-off planes. This signal is called LPCPD# on the LPC interface. |
| SUSCLK | O | **Suspend Clock:** This clock is an output of the RTC generator circuit to use by other chips for refresh clock. |
| VRMPWRGD | I | **VRM Power Good:** This should be connected to be the processor's VRM Power Good signifying the VRM is stable. This signal is internally ANDed with the PWROK input. |
| BM_BUSY#/ GPIO0 | I | **Bus Master Busy:** This signal supports the C3 state. It provides an indication that a bus master device is busy. When this signal is asserted, the BM_STS bit will be set. If this signal goes active in a C3 state, it is treated as a break event.<br>**NOTE:** This signal is internally synchronized using the PCICLK and a two-stage synchronizer. It does not need to meet any particular setup or hold time. |
| CLKRUN# | I/O | **PCI Clock Run:** This clock supports the PCI CLKRUN protocol. It connects to peripherals that need to request clock restart or prevention of clock stopping. |
| STP_PCI# | O | **Stop PCI Clock:** This signal is an output to the external clock generator for it to turn off the PCI clock. It is used to support PCI CLKRUN# protocol. If this functionality is not needed, this signal can be configured as a GPIO. |
| STP_CPU# | O | **Stop CPU Clock:** This signal is an output to the external clock generator for it to turn off the processor clock. It is used to support the C3 state. If this functionality is not needed, this signal can be configured as a GPIO. |

**Table 2-13.Power Management Interface Signals (Sheet 3 of 3)**

| Name | Type | Description |
|---|---|---|
| BATLOW# | I | **Battery Low:** This signal is an input from battery to indicate that there is insufficient power to boot the system. Assertion will prevent wake from S3–S5 state. This signal can also be enabled to cause an SMI# when asserted. |
| DPRSLPVR | O | **Deeper Sleep - Voltage Regulator:** This signal is used to lower the voltage of VRM during the C4 state. When the signal is high, the voltage regulator outputs the lower “Deeper Sleep” voltage. When low (default), the voltage regulator outputs the higher “Normal” voltage. |
| **DPRSTP#** | O | **Deeper Stop:** This is a copy of the DPRSLPVR and it is active low. |

## 2.12 Processor Interface

**Table 2-14.Processor Interface Signals (Sheet 1 of 2)**

| Name | Type | Description |
|---|---|---|
| A20M# | O | **Mask A20:** A20M# will go active based on either setting the appropriate bit in the Port 92h register, or based on the A20GATE input being active. |
| CPUSLP# | O | **CPU Sleep:** This signal puts the processor into a state that saves substantial power compared to Stop-Grant state. However, during that time, no snoops occur. The chipset can optionally assert the CPUSLP# signal when going to the S1 state. |
| FERR# | I | **Numeric Coprocessor Error:** This signal is tied to the coprocessor error signal on the processor. FERR# is only used if the chipset co-processor error reporting function is enabled in the OIC.CEN register (Chipset Config Registers:Offset 31FFh: bit 1). If FERR# is asserted, The chipset generates an internal IRQ13 to its interrupt controller unit. It is also used to gate the IGNNE# signal to ensure that IGNNE# is not asserted to the processor unless FERR# is active. FERR# requires an external weak pull-up to ensure a high level when the coprocessor error function is disabled.<br>**NOTE:** FERR# can be used in some states for notification by the processor of pending interrupt events. This functionality is independent of the OIC register bit setting. |
| IGNNE# | O | **Ignore Numeric Error:** This signal is connected to the ignore error pin on the processor. IGNNE# is only used if The chipset co-processor error reporting function is enabled in the OIC.CEN register (Chipset Config Registers:Offset 31FFh: bit 1). If FERR# is active, indicating a coprocessor error, a write to the Coprocessor Error register (I/O register F0h) causes the IGNNE# to be asserted. IGNNE# remains asserted until FERR# is negated. If FERR# is not asserted when the Coprocessor Error register is written, the IGNNE# signal is not asserted. |
| INIT# | O | **Initialization:** INIT# is asserted by The chipset for 16 PCI clocks to reset the processor. The chipset can be configured to support processor Built In Self Test (BIST). |
| INIT3_3V# | O | **Initialization 3.3 V:** This is the identical 3.3 V copy of INIT# intended for Firmware Hub. |

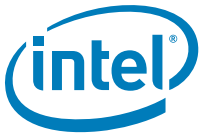


**Table 2-14.Processor Interface Signals (Sheet 2 of 2)**

| Name | Type | Description |
|---|---|---|
| INTR | O | **CPU Interrupt:** INTR is asserted by the chipset to signal the processor that an interrupt request is pending and needs to be serviced. It is an asynchronous output and normally driven low. |
| NMI | O | **Non-Maskable Interrupt:** NMI is used to force a non-Maskable interrupt to the processor. The chipset can generate an NMI when either SERR# is asserted or IOCHK# goes active via the SERIRQ# stream. The processor detects an NMI when it detects a rising edge on NMI. NMI is reset by setting the corresponding NMI source enable/disable bit in the NMI Status and Control register (I/O Register 61h). |
| SMI# | O | **System Management Interrupt:** SMI# is an active low output synchronous to PCICLK. It is asserted by the chipset in response to one of many enabled hardware or software events. |
| STPCLK# | O | **Stop Clock Request:** STPCLK# is an active low output synchronous to PCICLK. It is asserted by the chipset in response to one of many hardware or software events. When the processor samples STPCLK# asserted, it responds by stopping its internal clock. |
| RCIN# | I | **Keyboard Controller Reset CPU:** The keyboard controller can generate INIT# to the processor. This saves the external OR gate with the chipset's other sources of INIT#. When the chipset detects the assertion of this signal, INIT# is generated for 16 PCI clocks.<br>**NOTE:** The chipset will ignore RCIN# assertion during transitions to the S1, S3, S4, and S5 states. |
| A20GATE | I | **A20 Gate:** A20GATE is from the keyboard controller. The signal acts as an alternative method to force the A20M# signal active. It saves the external OR gate needed with various other chipsets. |
| CPUPWRGD / GPIO49 | O | **CPU Power Good:** This signal should be connected to the processor's PWRGOOD input to indicate when the CPU power is valid. This is an output signal that represents a logical AND of the chipset's PWROK and VRMPWRGD signals.<br>This signal may optionally be configured as a GPIO. |
| DPSLP# | O | **Deeper Sleep:** DPSLP# is asserted by the chipset to the processor. When the signal is low, the processor enters the deep sleep state by gating off the processor core clock inside the processor. When the signal is high (default), the processor is not in the deep sleep state. |

## 2.13 SMBus Interface

**Table 2-15.SM Bus Interface Signals**

| Name | Type | Description |
|---|---|---|
| SMBDATA | I/OD | **SMBus Data:** External pull-up resistor is required. |
| SMBCLK | I/OD | **SMBus Clock:** External pull-up resistor is required. |
| SMBALERT# / GPIO11 | I | **SMBus Alert:** This signal is used to wake the system or generate SMI#. If not used for SMBALERT#, it can be used as a GPIO. |

## 2.14 System Management Interface

**Table 2-16.System Management Interface Signals**

| Name | Type | Description |
|---|---|---|
| INTRUDER# | I | **Intruder Detect:** This signal can be set to disable the system if the chasis is detected open. This signal's status is readable, so it can be used like a GPIO if the Intruder Detection is not needed. |
| SMLINK[1:0] | I/OD | **System Management Link:** These signals provide a SMBus link to optional external system management ASIC or LAN controller. External pull-ups are required. Note that SMLINK0 corresponds to an SMBus clock signal, and SMLINK1 corresponds to an SMBus Data signal. |
| LINKALERT# | I/OD | **SMLink Alert:** This signal is an output of the integrated LAN and input to either the integrated ASF or an external management controller in order for the LAN's SMLINK slave to be serviced. |

## 2.15 Real Time Clock Interface

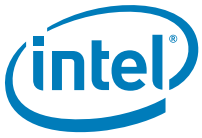
**Table 2-17.Real Time Clock Interface**

| Name | Type | Description |
|---|---|---|
| RTCX1 | Special | **Crystal Input 1:** This signal is connected to the 32.768 kHz crystal. If no external crystal is used, RTCX1 can be driven with the desired clock rate. |
| RTCX2 | Special | **Crystal Input 2:** This signal is connected to the 32.768 kHz crystal. If no external crystal is used, RTCX2 should be left floating. |

## 2.16 Other Clocks

**Table 2-18.Other Clocks**

| Name | Type | Description |
|---|---|---|
| CLK14 | I | **Oscillator Clock:** This clock signal is used for the 8254 timers. It runs at 14.31818 MHz. This clock is permitted to stop during S3 (or lower) states. |
| CLK48 | I | **48 MHz Clock:** This clock signal is used to run the USB controller. It runs at 48.000 MHz. This clock is permitted to stop during S3 (or lower) states. |
| SATA_CLKP<br>SATA_CLKN | I | 100 MHz Differential Clock: These signals are used to run the SATA controller at 100 MHz. This clock is permitted to stop during S3/S4/S5 states. |
| DMI_CLKP,<br>DMI_CLKN | I | **100 MHz Differential Clock:** These signals are used to run the Direct Media Interface. They run at 100 MHz. |



## 2.17 Miscellaneous Signals

**Table 2-19.Miscellaneous Signals**

| Name | Type | Description |
|---|---|---|
| INTVRMEN | I | **Internal Voltage Regulator Enable:** This signal must always connected to VccRTC to make sure the internal 1.05 V Suspend regulator is been enabled. |
| SPKR | O | **Speaker:** The SPKR signal is the output of counter 2 and is internally "ANDed" with Port 61h bit 1 to provide Speaker Data Enable. This signal drives an external speaker driver device, which in turn drives the system speaker. Upon PLTRST#, its output state is 0.<br>**NOTE:** SPKR is sampled at the rising edge of PWROK as a functional strap. See Section 2.22.1 for more details. There is a weak integrated pull-down resistor on SPKR pin. |
| RTCRST# | I | **RTC Reset:** When asserted, this signal resets register bits in the RTC well.<br><br>**NOTES:**<br>1. Unless CMOS is being cleared (only to be done in the G3 power state), the RTCRST# input must be high when all other RTC power planes are on.<br>2. In the case where the RTC battery is dead or missing on the platform, the RTCRST# pin must rise before the RSMRST# pin. |
| STRAP0 | O | **Strapping 0:** This pin have strapping function use as "Top-Block Swap Override". |
| STRAP[1]#/ GPIO48 | O | **Strapping 1:** This pin have strapping function use as "Boot BIOS Destination Selection".<br>The STRAP[1]# pin can instead be used as a GPIO. |
| STRAP[2]#/ GPIO17 | O | **Strapping 2:** This pin have strapping function use as "Boot BIOS Destination Selection".<br>The STRAP[2]# pin can instead be used as a GPIO. |

## 2.18 Intel HD Audio Link

**Table 2-20.Intel HD Audio Link Signals (Sheet 1 of 2)**

| Name[1,2] | Type | Description |
|---|---|---|
| HDA_RST# | O | **Intel HD Audio Reset:** This signal is the master hardware reset to external codec(s). |
| HDA_SYNC | O | **Intel High Definition Audio Sync:** This signal is a 48 kHz fixed rate sample sync to the codec(s). It is also used to encode the stream number. |
| HDA_BIT_CLK | O | **Intel High Definition Audio Bit Clock Output:** This signal is a 24.000 MHz serial data clock generated by the Intel High Definition Audio controller. This signal has an integrated pull-down resistor so that HDA_BIT_CLK doesn't float when an Intel High Definition Audio codec (or no codec) is connected. |

**Table 2-20.Intel HD Audio Link Signals (Sheet 2 of 2)**

| Name[1,2] | Type | Description |
|---|---|---|
| HDA_SDOUT | O | **Intel High Definition Audio Serial Data Out:** This signal is the serial TDM data output to the codec(s). This serial output is double-pumped for a bit rate of 48 Mb/s for Intel High Definition Audio. **NOTE:** HDA_SDOUT is sampled at the rising edge of PWROK as a functional strap. See Section 2.22.1 for more details. There is a weak integrated pull-down resistor on the HDA_SDOUT pin. |
| HDA_SDIN[2:0] | I | **Intel High Definition Audio Serial Data In [2:0]:** These signals are serial TDM data inputs from the three codecs. The serial input is single-pumped for a bit rate of 24 Mb/s for Intel HD Audio. These signals have integrated pull-down resistors that are always enabled. |

1. Some signals have integrated pull-ups or pull-downs. Consult table in Section 3.1 for details.
2. Intel High Definition Audio have to configure through D30:F1:40h, bit 0: AZ#. This bit configure the Intel High Definition Audio signals and BIOS need to set it to 1.

## 2.19 Serial Peripheral Interface (SPI)

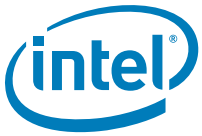
**Table 2-21.Serial Peripheral Interface (SPI) Signals**

| Name | Type | Description |
|---|---|---|
| SPI_CS# | I/O | **SPI Chip Select:** This chip select signal is also used as the SPI bus request signal. |
| SPI_MISO | I | **SPI Master IN Slave OUT**: This signal is the data input pin for the chipset. |
| SPI_MOSI | O | **SPI Master OUT Slave IN**: This signal is the data output pin for the chipset. |
| SPI_ARB | I | **SPI Arbitration**: SPI_ARB is the SPI arbitration signal used to arbitrate the SPI bus when Shared Flash is implemented. |
| SPI_CLK | O | **SPI Clock**: This signal is the SPI clock signal. During idle, the bus owner will drive the clock signal low. 17.86 MHz. |

## 2.20 General Purpose I/O Signals

**Table 2-22.General Purpose I/O Signals (Sheet 1 of 2)**

| Name[1,2] | Type | Tolerance | Power Well | Default | Description |
|---|---|---|---|---|---|
| GPIO49 | I/O | V_CPU_IO | V_CPU_IO | Native | Multiplexed with CPUPWRGD |
| GPIO48 | I/O | 3.3 V | Core | Native | Multiplexed with STRAP1# |
| GPIO[47:40] | N/A | N/A | N/A | N/A | Not implemented. |
| GPIO[39:38] | I/O | 3.3 V | Core | GPI | Unmultiplexed. |
| GPIO37 | N/A | N/A | N/A | N/A | Not Implemented. |
| GPIO36 | I/O | 3.3 V | Core | GPI | Unmultiplexed. |



### Table 2-22.General Purpose I/O Signals (Sheet 2 of 2)

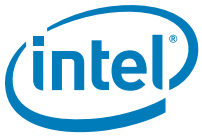
| Name[1,2] | Type | Tolerance | Power Well | Default | Description |
|---|---|---|---|---|---|
| GPIO35 | N/A | N/A | N/A | N/A | Not Implemented. |
| GPIO34 | I/O | 3.3 V | Core | GPO | Unmultiplexed. |
| GPIO33 | I/O | 3.3 V | Core | GPO | Unmultiplexed. |
| GPIO32 | N/A | N/A | N/A | N/A | Not Implemented. |
| GPIO31 | I/O | 3.3 V | Resume | Native | Multiplexed with OC7# |
| GPIO30 | I/O | 3.3 V | Resume | Native | Multiplexed with OC6# |
| GPIO29 | I/O | 3.3 V | Resume | Native | Multiplexed with OC5# |
| GPIO28 | I/O | 3.3 V | Resume | GPO | Unmultiplexed. |
| GPIO27 | I/O | 3.3 V | Resume | GPO | Unmultiplexed. |
| GPIO26 | I/O | 3.3 V | Resume | GPO | Unmultiplexed. |
| GPIO25 | I/O | 3.3 V | Resume | GPO | Unmultiplexed. |
| GPIO24 | I/O | 3.3 V | Resume | GPO | Unmultiplexed. Not cleared by CF9h reset event. |
| GPIO23 | I/O | 3.3 V | Core | Native | Multiplexed with LDRQ1# |
| GPIO22 | I/O | 3.3 V | Core | GPI | Unmultiplexed. |
| GPIO[21:18] | N/A | N/A | N/A | N/A | Not Implemented |
| GPIO17 | I/O | 3.3 V | Core | GPO | Multiplexed with STRAP2#. |
| GPIO16 | N/A | N/A | N/A | N/A | Not Implemented |
| GPIO[15:12] | I/O | 3.3 V | Resume | GPI | Unmultiplexed. |
| GPIO11 | I/O | 3.3 V | Resume | Native | Multiplexed with SMBALERT# |
| GPIO[10:8] | I/O | 3.3 V | Resume | GPI | Unmultiplexed. |
| GPIO[7:6] | I/O | 3.3 V | Core | GPI | Unmultiplexed. |
| GPIO[5:2] | I/OD | 5 V | Core | GPI | Multiplexed with PIRQ[H:E]#. |
| GPIO1 | I/O | 5 V | Core | GPI | Unmultiplexed. |
| GPIO0 | I/O | 3.3 V | Core | GPI | Multiplexed with BM_BUSY#. |

1. GPI[15:0] can be configured to cause a SMI# or SCI. Note that a GPI can be routed to either an SMI# or an SCI, but not both.
2. Some GPIOs exist in the VccSus3_3 power plane. Care must be taken to make sure GPIO signals are not driven high into powered-down planes. Some chipset GPIOs may be connected to pins on devices that exist in the core well. If these GPIOs are outputs, there is a danger that a loss of core power (PWROK low) or a Power Button Override event will result in the Intel chipset driving a pin to a logic 1 to another device that is powered down.

## 2.21 Power and Ground

**Table 2-23.Power and Ground Signals**

| Name | Description |
|---|---|
| Vcc3_3 | These pins provide the 3.3 V supply for core well I/O buffers (6pins). This power may be shut off in S3, S4, S5 or G3 states. |
| Vcc1_05 | These pins provide the 1.05 V supply for core well logic (4 pins). This power may be shut off in S3, S4, S5 or G3 states. |
| Vcc1_5 | These pins provide the 1.5 V supply for Logic and I/O (4 pins). This power may be shut off in S3, S4, S5 or G3 states. |
| V5REF | These pins provide the reference for 5 V tolerance on core well inputs (1 pins). This power may be shut off in S3, S4, S5 or G3 states. |
| VccSus3_3 | These pins provide the 3.3 V supply for resume well I/O buffers (4 pins). This power is not expected to be shut off unless the system is unplugged in Nettop configurations or the main battery is removed or completely drained and AC power is not available in Netbook configurations. |
| V5REF_Sus | This pin provides the reference for 5 V tolerance on resume well inputs (1 pin). This power is not expected to be shut off unless the system is unplugged in Nettop configurations or the main battery is removed or completely drained and AC power is not available in Netbook configurations. |
| VccRTC | This pin provides the 3.3 V (can drop to 2.0 V min. in G3 state) supply for the RTC well (1 pin). This power is not expected to be shut off unless the RTC battery is removed or completely drained.<br><br>**NOTE:** Implementations should not attempt to clear CMOS by using a jumper to pull VccRTC low. Clearing CMOS in a chipset-based platform can be done by using a jumper on RTCRST# or GPI. |
| VccUSBPLL | This pin provides the 1.5 V supply for core well logic (1 pin). This signal is used for the USB PLL. This power may be shut off in S3, S4, S5 or G3 states. Must be powered even if USB not used. |
| VccDMIPLL | This pin provides the 1.5 V supply for core well logic (1 pin). This signal is used for the DMI PLL. This power may be shut off in S3, S4, S5 or G3 states. |
| VccSATAPLL | This pin provides the 1.5 V supply for core well logic (1 pin). This signal is used for the SATA PLL. This power may be shut off in S3, S4, S5 or G3 states. Must be powered even if SATA not used. |
| V_CPU_IO | These pins are powered by the same supply as the processor I/O voltage (1 pins). This supply is used to drive the processor interface signals listed in Table 2-14. |
| Vss | Grounds (59 pins). |



## 2.22 Pin Straps

### 2.22.1 Functional Straps

The following signals are used for static configuration. They are sampled at the rising edge of PWROK to select configurations (except as noted), and then revert later to their normal usage. To invoke the associated mode, the signal should be driven at least four PCI clocks prior to the time it is sampled.

**Table 2-24.Functional Strap Definitions (Sheet 1 of 2)**

| Signal | Usage | When Sampled | Comment |
|---|---|---|---|
| HDA_SDOUT | PCI Express* Port Config bit 1 | Rising Edge of PWROK | When TP3 not pulled low at rising edge of PWROK, sets bit 1 of RPC.PC (Chipset Configuration Registers:Offset 224h). See Section 10.1.30 for details.<br>This signal has a weak internal pull-down. |
| HDA_SYNC | PCI Express Port Config bit 0 | Rising Edge of PWROK | This signal has a weak internal pull-down.<br>Sets bit 0 of RPC.PC (Chipset Configuration Registers:Offset 224h). See Section 10.1.30 for details. |
| EE_CS | Reserved | | This signal has a weak internal pull-down.<br>**NOTE:** This signal should not be pulled high. |
| EE_DOUT | Reserved | | This signal has a weak internal pull-up.<br>**NOTE:** This signal should not be pulled low. |
| GNT2# | Reserved | | This signal has a weak internal pull-up.<br>**NOTE:** This signal should not be pulled low. |
| STRAP0# | Top-Block Swap Override | Rising Edge of PWROK | The signal has a weak internal pull-up. If the signal is sampled low, this indicates that the system is strapped to the “top-block swap” mode (the chipset inverts A16 for all cycles targeting FWH BIOS space). The status of this strap is readable via the Top Swap bit (Chipset Configuration Registers:Offset 3414h:bit 0). Note that software will not be able to clear the Top-Swap bit until the system is rebooted without STRAP0# being pulled down. |
| STRAP2# / GPIO17, STRAP1# / GPIO48 | Boot BIOS Destination Selection | Rising Edge of PWROK | This field determines the destination of accesses to the BIOS memory range. Signals have weak internal pull-ups. Also controllable via Boot BIOS Destination bit (Chipset Configuration Registers:Offset 3410h:bit 11:10)<br>(STRAP2# is MSB)<br>01 = SPI<br>10 = PCI<br>11 = LPC |
| DPRSLPVR | Reserved | | This signal has a weak internal pull-down.<br>**NOTE:** This signal should not be pulled high. |

**Table 2-24.Functional Strap Definitions (Sheet 2 of 2)**

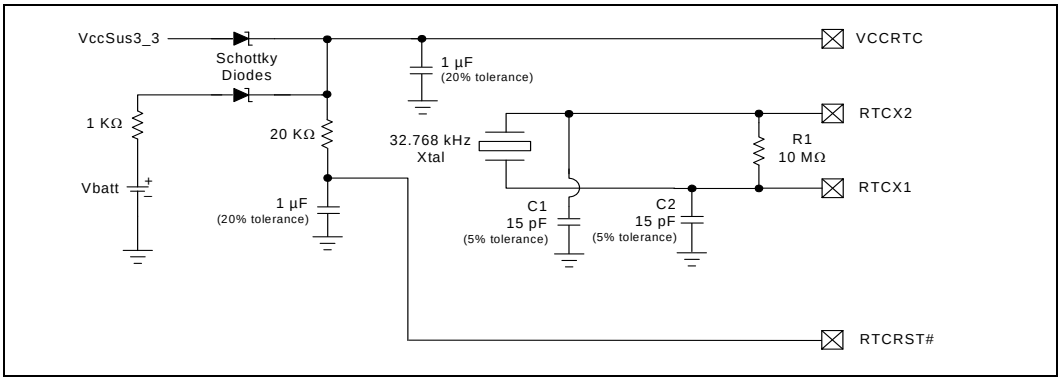
| Signal | Usage | When Sampled | Comment |
|---|---|---|---|
| GPIO25 | DMI AC/DC Coupling Selection | Rising Edge of RSMRST# | This signal has a weak internal pull-up.The internal pull-up is disabled within 100 ms after RSMRST# deasserts.<br>If the signal is sampled high, the DMI interface is strapped to operate in DC coupled mode (No coupling capacitors are required on DMI differential pairs).<br>If the signal is sampled low, the DMI interface is strapped to operate in AC coupled mode (Coupling capacitors are required on DMI differential pairs).<br>**NOTES:**<br>1. Board designer must ensure that DMI implementation matches the strap selection.<br>2. The signal must be held low at least 2 us after RSMRST# deassertion to enable AC coupled mode.<br>3. When the chipset pairs with Intel® Atom™ Processor D400, D500 and N400 series, should enable AC couple mode for Receive signal and DC couple mode for Transmit signal. |
| INTVRMEN | Enable Integrated VccSus1_05 VRM | Always | Enables integrated VccSus1_05 VRM when sampled high. |
| LINKALERT# | Reserved | | This signal requires an external pull-up resistor. |
| SATALED# | Reserved | | This signal has a weak internal pull-up enabled only when PLTRST# is asserted.<br>**NOTE:** This signal should not be pulled low. |
| SPKR | No Reboot | Rising Edge of PWROK | The signal has a weak internal pull-down. If the signal is sampled high, this indicates that the system is strapped to the “No Reboot” mode (the chipset will disable the TCO Timer system reboot feature). The status of this strap is readable via the NO REBOOT bit (Chipset Config Registers:Offset 3410h:bit 5). |

**NOTE:** See Section 3.1for full details on pull-up/pull-down resistors.

## 2.22.2 External RTC Circuitry

To reduce RTC well power consumption, the chipset implements an internal oscillator circuit that is sensitive to step voltage changes in VccRTC. Following figure shows an example schematic recommended to ensure correct operation of chipset RTC.

**Figure 2-3. Example External RTC Circuit**




**NOTE:** C1 and C2 depend on crystal load.

## 2.23 Device and Revision ID Table

| Device Function | Description | Chipset Dev ID[1] | Chipset Rev ID | Comments |
|---|---|---|---|---|
| D31, F0 | LPC | 27BCh | 00h | |
| D31, F2 | SATA | 27C0h | 00h | Non-AHCI Mode[1] |
| | | 27C1h | 00h | AHCI Mode[1] |
| D31, F3 | SMBus | 27DAh | 00h | |
| D30, F0 | DMI to PCI Bridge | 2448h | E0h | |
| D29, F0 | USB UHC #1 | 27C8h | 00h | |
| D29, F1 | USB UHC #2 | 27C9h | 00h | |
| D29, F2 | USB UHC #3 | 27CAh | 00h | |
| D29, F3 | USB UHC #4 | 27CBh | 00h | |
| D29, F7 | USB EHCI | 27CCh | 00h | |
| D28:F0 | PCI Express* Port 1 | 27D0 | 00h | |
| D28:F1 | PCI Express Port 2 | 27D2 | 00h | |
| D28:F2 | PCI Express Port 3 | 27D4 | 00h | |
| D28:F3 | PCI Express Port 4 | 27D6 | 00h | |
| D27:F0 | Intel HD Audio | 27D8 | 00h | |
| D8: F0 | LAN | See Note 2 | 00h | |

**NOTES:**
1. Intel NM10 contains a single SATA device. The SATA Device ID is dependant upon which SATA mode is selected by BIOS and capabilities exist in the SKU.
2. Loaded from EEPROM. If EEPROM contains either 0000h or FFFFh in the device ID location, then 27DCh is used. Refer to the ICH7 EEPROM Map and Programming Guide for LAN Device IDs.§

§

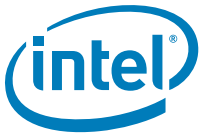
# 3 Pin States

## 3.1 Integrated Pull-Ups and Pull-Downs

**Table 3-25. Integrated Pull-Up and Pull-Down Resistors**

| Signal | Resistor | Nominal | Notes |
|---|---|---|---|
| HDA_BIT_CLK, Intel High Definition Audio | Pull-Down | 20 kΩ | 2, 6, 7 |
| HDA_RST#, Intel High Definition Audio | None | N/A | 2 |
| HDA_SDIN[2:0], Intel High Definition Audio | Pull-down | 20 kΩ | 2, 4 |
| HDA_SDOUT, Intel High Definition Audio | Pull-down | 20 kΩ | 1, 2 |
| HDA_SYNC, Intel High Definition Audio | Pull-down | 20 kΩ | 2, 4 |
| DPRSLPVR | Pull-down | 20 kΩ | 4, 9 |
| EE_CS | Pull-down | 20 kΩ | 10, 11 |
| EE_DIN | Pull-up | 20 kΩ | 10 |
| EE_DOUT | Pull-up | 20 kΩ | 10 |
| GNT1# | Pull-up | 20 kΩ | 10, 12 |
| GNT2# | Pull-up | 20 kΩ | 10, 19 |
| STRAP0#<br>STRAP1#/ GPIO48<br>STRAP2#/ GPIO17 | Pull-up | 20 kΩ | 10, 19 |
| GPIO25 | Pull-up | 20 kΩ | 10, 13 |
| LAD[3:0] / FHW[3:0]# | Pull-up | 20 kΩ | 10 |
| LAN_CLK | Pull-down | 100 kΩ | 14 |
| LAN_RXD[2:0] ( | Pull-up | 20 kΩ | 15 |
| LDRQ0# | Pull-up | 20 kΩ | 10 |
| LDRQ1# / GPIO23 | Pull-up | 20 kΩ | 10 |
| PME# | Pull-up | 20 kΩ | 10 |
| PWRBTN# | Pull-up | 20 kΩ | 10 |
| SATALED# | Pull-up | 15 kΩ | 16 |
| SPI_ARB | Pull-down | 20 kΩ | 10 |
| SPI_CLK | Pull-down | 20 kΩ | 10 |
| SPKR | Pull-down | 20 kΩ | 4 |
| USBP[7:0] [P,N] | Pull-down | 15 kΩ | 18 |

**NOTES:**See next page



1. The pull-down resistors on HDA_BIT_CLK (AC '97) and HDA_RST# are enabled when either:
   - The LSO bit (bit 3) in the AC'97 Global Control Register (D30:F2:2C) is set to 1, or
   - Both Function 2 and Function 3 of Device 30 are disabled.
   - Otherwise, the integrated Pull-down resistor is disabled.
2. The Intel High Definition Audio Link signals must be configured to be an Intel High Definition Audio Link.
3. Simulation data shows that these resistor values can range from 10 kΩ to 20 kΩ.
4. Simulation data shows that these resistor values can range from 9 kΩ to 50 kΩ.
5. Simulation data shows that these resistor values can range from 10 kΩ to 40 kΩ.
6. The pull-down on this signal (in Intel High Definition Audio mode) is only enabled when in $S3_{COLD}$.
7. Simulation data shows that these resistor values can range from 5.7 kΩ to 28.3 kΩ.
8. The pull-up or pull-down on this signal is only enabled at boot/reset for strapping function.
9. Simulation data shows that these resistor values can range from 15 kΩ to 35 kΩ.
10. The pull-down on this signal is only enabled when LAN_RST# is asserted.
11. The internal pull-up is enabled only when the PCIRST# pin is driven low and the PWROK indication is high.
12. Internal pull-up is enabled during RSMRST# and is disabled within 100 ms after RSMRST# de-asserts.
13. Simulation data shows that these resistor values can range from 45 kΩ to 170 kΩ.
14. Simulation data shows that these resistor values can range from 15 kΩ to 30 kΩ.
15. Simulation data shows that these resistor values can range from 10 kΩ to 20 kΩ. The internal pull-up is only enabled only during PLTRST# assertion.
16. Simulation data shows that these resistor values can range from 10 kΩ to 30 kΩ.
17. Simulation data shows that these resistor values can range from 14.25 kΩ to 24.8 kΩ.
18. The internal pull-up is enabled only when PCIRST# is low.

## 3.2 Output and I/O Signals Planes and States

Table 3-26 shows the power plane associated with the output and I/O signals, as well as the state at various times. Within the table, the following terms are used:

| | |
|---|---|
| "High-Z" | Tri-state. the chipset not driving the signal high or low. |
| "High" | The chipset is driving the signal to a logic 1 |
| "Low" | The chipset is driving the signal to a logic 0 |
| "Defined" | Driven to a level that is defined by the function (will be high or low) |
| "Undefined" | The chipset is driving the signal, but the value is indeterminate. |
| "Running" | Clock is toggling or signal is transitioning because function not stopping |
| "Off" | The power plane is off, so the chipset is not driving |

Note that the signal levels are the same in S4 and S5, except as noted.

The chipset suspend well signal states are indeterminate and undefined and may glitch, including input signals acting as outputs, prior to RSMRSTB deassertion. This does not apply to LAN_RST#, SLP_S3#, SLP_S4#, and SLP_S5#. These signals are determinate and defined prior to RSMRST# deassertion.

The chipset core well signal states are indeterminate and undefined and may glitch, including input signals acting as outputs, prior to PWROK assertion. This does not apply to FERR# and THRMTRIP#. These signals are determinate and defined prior to PWROK assertion.

**Table 3-26.Power Plane and States for Output and I/O Signals (Sheet 1 of 4)**

| Signal Name | Power Plane | During PLTRST#[6] / RSMRST#[7] | Immediately after PLTRST#[6] / RSMRST#[7] | C3/C4 | S1 | $S3_{COLD}$[13] | S4/S5 |
|---|---|---|---|---|---|---|---|
| **PCI Express*** | | | | | | | |
| PETp[4:1], PETn[4:1] | Core | High | High[12] | Defined | Defined | Off | Off |
| **DMI** | | | | | | | |
| DMI[3:0]TXP, DMI[3:0]TXN | Core | High | High[12] | Defined | Defined | Off | Off |
| **PCI Bus** | | | | | | | |
| AD[31:0] | Core | Low | Undefined | Defined | Defined | Off | Off |
| C/BE[3:0]# | Core | Low | Undefined | Defined | Defined | Off | Off |
| DEVSEL# | Core | High-Z | High-Z | High-Z | High-Z | Off | Off |
| FRAME# | Core | High-Z | High-Z | High-Z | High-Z | Off | Off |
| GNT[2:1]# STRAP1#/GPIO48 STRAP2#/GPIO17 | Core | High with Internal Pull-ups | High | High | High | Off | Off |
| IRDY#, TRDY# | Core | High-Z | High-Z | High-Z | High-Z | Off | Off |
| PAR | Core | Low | Undefined | Defined | Defined | Off | Off |
| PCIRST# | Suspend | Low | High | High | High | Low | Low |
| PERR# | Core | High-Z | High-Z | High-Z | High-Z | Off | Off |
| PLOCK# | Core | High-Z | High-Z | High-Z | High-Z | Off | Off |
| STOP# | Core | High-Z | High-Z | High-Z | High-Z | Off | Off |
| **LPC Interface** | | | | | | | |
| LAD[3:0] / FWH[3:0] | Core | High | High | High | High | Off | Off |
| LFRAME# | Core | High | High | High | High | Off | Off |
| **LAN Connect and EEPROM Interface** | | | | | | | |
| EE_CS | LAN | Low | Running | Defined | Defined | Note 4 | Note 4 |
| EE_DOUT | LAN | High | High | Defined | Defined | Note 4 | Note 4 |
| EE_SHCLK | LAN | High-Z | Running | Defined | Defined | Note 4 | Note 4 |
| LAN_RSTSYNC | LAN | High | Low | Defined | Defined | Note 4 | Note 4 |
| LAN_TXD[2:0] | LAN | Low | Low | Defined | Defined | Note 4 | Note 4 |

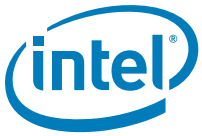


**Table 3-26.Power Plane and States for Output and I/O Signals (Sheet 2 of 4)**

| Signal Name | Power Plane | During PLTRST#[6] / RSMRST#[7] | Immediately after PLTRST#[6] / RSMRST#[7] | C3/C4 | S1 | $S3_{COLD}$[13] | S4/S5 |
|---|---|---|---|---|---|---|---|
| **SATA Interface** | | | | | | | |
| SATA[0]TXP, SATA[0]TXN SATA[1]TXP, SATA[1]TXN | Core | High-Z | High-Z | Defined | Defined | Off | Off |
| SATALED# | Core | High-Z | High-Z | Defined | Defined | Off | Off |
| SATARBIAS | Core | High-Z | High-Z | Defined | Defined | Off | Off |
| **Interrupts** | | | | | | | |
| PIRQ[A:D]#, PIRQ[H:E]# / GPIO[5:2] | Core | High-Z | High-Z | Defined | High-Z | Off | Off |
| SERIRQ | Core | High-Z | High-Z | Running | High-Z | Off | Off |
| **USB Interface** | | | | | | | |
| USBP[7:0][P,N] | Suspend | Low | Low | Low | Low | Low | Low |
| USBRBIAS | Suspend | High-Z | High-Z | Defined | Defined | Defined | Defined |
| OC[7:5]#/ GP[31:29] | Suspend | Input | Input | Driven | Driven | Driven | Driven |
| **Power Management** | | | | | | | |
| PLTRST# | Suspend | Low | High | High | High | Low | Low |
| SLP_S3# | Suspend | Low | High | High | High | Low | Low |
| SLP_S4# | Suspend | Low | High | High | High | High | Low |
| SLP_S5# | Suspend | Low | High | High | High | High | Low[10] |
| STP_PCI# | Core | High | High | Defined | High | Low | Low |
| STP_CPU# | Core | High | High | Low | High | Low | Low |
| SUS_STAT# / LPCPD# | Suspend | Low | High | High | High | Low | Low |
| DPRSLPVR | Core | Low | Low | Low/ High[5] | High | Off | Off |
| DPRSTP# | Core | High | High | Low/ High[5] | High | Off | Off |
| SUSCLK | Suspend | Low | Running | | | | |

**Table 3-26.Power Plane and States for Output and I/O Signals (Sheet 3 of 4)**

| Signal Name | Power Plane | During PLTRST#[6] / RSMRST#[7] | Immediately after PLTRST#[6] / RSMRST#[7] | C3/C4 | S1 | $S3_{COLD}$[13] | S4/S5 |
|---|---|---|---|---|---|---|---|
| **Processor Interface** | | | | | | | |
| A20M# | Core | Dependant on A20GATE Signal | See Note 8 | Defined | High | Off | Off |
| CPUPWRGD / GPIO49 | Core | See Note 3 | High | High | High | Off | Off |
| IGNNE# | Core | High | See Note 8 | High | High | Off | Off |
| INIT# | Core | High | High | High | High | Off | Off |
| INIT3_3V# | Core | High | High | High | High | Off | Off |
| INTR | Core | See Note 8 | See Note 8 | Defined | Low | Off | Off |
| NMI | Core | See Note 8 | See Note 8 | Defined | Low | Off | Off |
| SMI# | Core | High | High | Defined | High | Off | Off |
| STPCLK# | Core | High | High | Low | Low | Off | Off |
| DPSLP# | Core | High | High | High/ Low | High | Off | Off |
| **SMBus Interface** | | | | | | | |
| SMBCLK, SMBDATA | Suspend | High-Z | High-Z | Defined | Defined | Defined | Defined |
| **System Management Interface** | | | | | | | |
| SMLINK[1:0] | Suspend | High-Z | High-Z | Defined | Defined | Defined | Defined |
| LINKALERT# | Suspend | High-Z | High-Z | Defined | Defined | Defined | Defined |
| **Miscellaneous Signals** | | | | | | | |
| SPKR | Core | High-Z with Internal Pull-down | Low | Defined | Defined | Off | Off |
| **Intel HD Audio Interface** | | | | | | | |
| HDA_RST# | Suspend | Low | Low[11] | High | TBD | Low | Low |
| HDA_SDOUT | Core | High-Z with Internal Pull-down | Running | Running | Low | Off | Off |
| HDA_SYNC | Core | High-Z with Internal Pull-down | Running | Running | Low | Off | Off |
| HDA_BIT_CLK | Core | High-Z with Internal Pull-down | Low[11] | Running | Low | Off | Off |

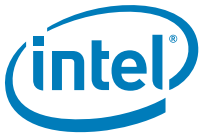


**Table 3-26.Power Plane and States for Output and I/O Signals (Sheet 4 of 4)**

| Signal Name | Power Plane | During PLTRST#[6] / RSMRST#[7] | Immediately after PLTRST#[6] / RSMRST#[7] | C3/C4 | S1 | $S3_{COLD}$[13] | S4/S5 |
|---|---|---|---|---|---|---|---|
| **Un-multiplexed GPIO Signals** | | | | | | | |
| GPIO[7:6] | Core | Input | Input | Driven | Driven | Off | Off |
| GPIO[15:12,10:8] | Suspend | Input | Input | Driven | Driven | Driven | Driven |
| GPIO24 | Suspend | No Change | No Change | Defined | Defined | Defined | Defined |
| GPIO25 | Suspend | High | High[7] | Defined | Defined | Defined | Defined |
| GPIO[28:26] | Suspend | Low | Low | Defined | Defined | Defined | Defined |
| GPIO33 | Suspend | High | High | Defined | Defined | Off | Off |
| GPIO34 | Core | Low | Low[11] | Defined | Defined | Off | Off |
| GPIO[39:38] | Core | Input | Input | Driven | Driven | Off | Off |
| **SPI Interface** | | | | | | | |
| SPI_CS# | Suspend | High | High | High | High | High | High |
| SPI_MOSI | Suspend | High | High | High | High | High | High |
| SPI_ARB | Suspend | Low | Low | Low | Low | Low | Low |
| SPI_CLK | Suspend | Low | Low | Low | Low | Low | Low |

**NOTES:**
1. NM10 drives these signals High after the CPU Reset.
2. GPIO[18] will toggle at a frequency of approximately 1 Hz when the NM10 comes out of reset
3. CPUPWRGD is an output that represents a logical AND of the NM10's VRMPWRGD and PWROK signals, and thus will be driven low by NM10 when either VRMPWRGD or PWROK are inactive. During boot, or during a hard reset with power cycling, CPUPWRGD will be expected to transition from low to High.
4. LAN Connect and EEPROM signals will either be "Defined" or "Off" in S3-S5 states depending upon whether or not the LAN power planes are active.
5. The state of the DPRSLPVR and DPRSTP# signals in C4 are high if Deeper Sleep is enabled or low if it is disabled.
6. The states of Vcc3_3 signals are taken at the times during PLTRST# and Immediately after PLTRST#.
7. The states of VccSus3_3 signals are taken at the times during RSMRST# and Immediately after RSMRST#.
8. NM10 drives these signals Low before PWROK rising and Low after the CPU Reset.
9. GPIO[25] transitions from pulled high internally to actively driven within 100 ms of the deassertion of the RSMRST# pin.
10. 10.SLP_S5# signals will be high in the S4 state.
11. Low until Intel High Definition Audio Controller Reset bit set (D27:F0:Offset HDBAR+08h:bit 0), at which time ACZ_RST# will be High and ACZ_BIT_CLK will be Running.
12. PETp/n[6:1] high until port is enabled by software.
13. In S3hot, signal states are platform implementation specific, as some external components and interfaces may be powered when the NM10 is in the S3hot state.

## 3.3 Power Planes for Input Signals

Table 3-27 shows the power plane associated with each input signal, as well as what device drives the signal at various times. Valid states include:

High

Low

Static: Will be high or low, but will not change

Driven: Will be high or low, and is allowed to change

Running: For input clocks

The chipset suspend well signal states are indeterminate and undefined and may glitch prior to RSMRST#deassertion. This does not apply to LAN_RST#, SLP_S3#, SLP_S4# and SLP_S5#. These signals are determinate and defined prior to RSMRST# deassertion.

The chipset core well signal states are indeterminate and undefined and may glitch prior to PWROK assertion. This does not apply to FERR# and THRMTRIP#. These signals are determinate and defined prior to PWROK assertion.

**Table 3-27.Power Plane for Input Signals (Sheet 1 of 2)**

| Signal Name | Power Well | Driver During Reset | C3/C4 | S1 | $S3_{COLD}^{2}$ | S4/S5 |
|---|---|---|---|---|---|---|
| A20GATE | Core | External Microcontroller | Static | Static | Low | Low |
| HDA_SDIN[2:0] (Intel HD Audio Mode) | Suspend | Intel HD Audio Codec | Driven | Low | Low | Low |
| BM_BUSY# / GPIO0[2] | Core | Graphics Component [(G)MCH] | Driven | High | Low | Low |
| BATLOW# | Suspend | Power Supply | High | High | High | High |
| CLK14 | Core | Clock Generator | Running | Running | Low | Low |
| CLK48 | Core | Clock Generator | Running | Running | Low | Low |
| DMI_CLKP DMI_CLKN | Core | Clock Generator | Running | Running | Low | Low |
| EE_DIN | LAN | EEPROM Component | Driven | Driven | Note 1 | Note 1 |
| FERR# | Core | Processor | Static | Static | Low | Low |
| PERp[4:1], PERn[4:1] | Core | PCI Express* Device | Driven | Driven | Driven | Driven |
| DMI[3:0]RXP, DMI[3:0]RXN | Core | (G)MCH | Driven | Driven | Low | Low |
| INTRUDER# | RTC | External Switch | Driven | Driven | Driven | Driven |
| INTVRMEN | RTC | External Pull-up or Pull-down | Driven | Driven | Driven | Driven |

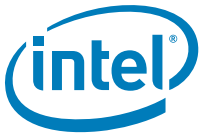


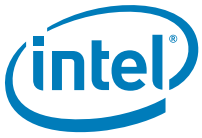
**Table 3-27.Power Plane for Input Signals (Sheet 2 of 2)**

| Signal Name | Power Well | Driver During Reset | C3/C4 | S1 | $S3_{COLD}$[2] | S4/S5 |
|---|---|---|---|---|---|---|
| LAN_CLK | LAN | LAN Connect Component | Driven | Driven | Note 1 | Note 1 |
| LAN_RST# | Suspend | Power Supply | High | High | Static | Static |
| LANRXD[2:0] | LAN | LAN Connect Component | Driven | Driven | Note 1 | Note 1 |
| LDRQ0# | Core | LPC Devices | Driven | High | Low | Low |
| LDRQ1# / GPIO23[3] | Core | LPC Devices | Driven | High | Low | Low |
| MCH_SYNC# | Core | (G)MCH | Driven | Driven | Low | Low |
| OC[7:0]# | Suspend | External Pull-ups | Driven | Driven | Driven | Driven |
| PCICLK | Core | Clock Generator | Running | Running | Low | Low |
| PME# | Suspend | Internal Pull-up | Driven | Driven | Driven | Driven |
| PWRBTN# | Suspend | Internal Pull-up | Driven | Driven | Driven | Driven |
| PWROK | RTC | System Power Supply | Driven | Driven | Low | Low |
| RCIN# | Core | External Microcontroller | High | High | Low | Low |
| REQ[2:1]#, | Core | PCI Master | Driven | Driven | Low | Low |
| RI# | Suspend | Serial Port Buffer | Driven | Driven | Driven | Driven |
| RSMRST# | RTC | External RC Circuit | High | High | High | High |
| RTCRST# | RTC | External RC Circuit | High | High | High | High |
| SATA_CLKP, SATA_CLKN | Core | Clock Generator | Running | Running | Low | Low |
| SATA[1:0]RXP, SATA[1:0]RXN | Core | SATA Drive | Driven | Driven | Driven | Driven |
| SATARBIAS# | Core | External Pull-Down | Driven | Driven | Driven | Driven |
| SERR# | Core | PCI Bus Peripherals | Driven | High | Low | Low |
| SMBALERT# / GPIO11[3] | Suspend | External Pull-up | Driven | Driven | Driven | Driven |
| SYS_RESET# | Suspend | External Circuit | Driven | Driven | Driven | Driven |
| THRM# | Core | Thermal Sensor | Driven | Driven | Low | Low |
| THRMTRIP# | Core | Thermal Sensor | Driven | Driven | Low | Low |
| USBRBIAS# | Suspend | External Pull-down | Driven | Driven | Driven | Driven |
| VRMPWRGD | Core | Processor Voltage Regulator | Driven | Driven | Low | Low |
| WAKE# | Suspend | External Pull-up | Driven | Driven | Driven | Driven |
| SPI_MISO | Suspend | External Pull-up | Driven | Driven | Driven | Driven |
| SPI_ARB | Suspend | Internal Pull-down | Low | Low | Low | Low |

**NOTES:**See next page

1. LAN Connect and EEPROM signals will either be “Driven” or “Low” in S3–S5 states depending upon whether or not the LAN power planes are active.
2. In S3hot, signal states are platform implementation specific, as some external components and interfaces may be powered when the NM10 is in the S3hot state.
3. These signals can be configured as outputs in GPIO mode.

§



# 4 Chipset and System Clock Domains

Table 4-28 shows the system clock domains. Figure 4-4 and Figure 4-5 show the assumed connection of the various system components, including the clock generator in Nettop and Netbook systems. For complete details of the system clocking solution, refer to the system's clock generator component specification.

**Table 4-28.Chipset and System Clock Domains**

| Clock Domain | Frequency | Source | Usage |
|---|---|---|---|
| Chipset SATA_CLKP, SATA_CLKN | 100 MHz | Main Clock Generator | Differential clock pair used for SATA. |
| Chipset DMI_CLKP, DMI_CLKN | 100 MHz | Main Clock Generator | Differential clock pair used for DMI. |
| Chipset PCICLK | 33 MHz | Main Clock Generator | Free-running PCI Clock to Chipset. This clock remains on during S0 and S1 (in Nettop) state, and is expected to be shut off during S3 or below in Nettop configurations or S1 or below in Netbook configurations. |
| System PCI | 33 MHz | Main Clock Generator | PCI Bus, LPC I/F. These only go to external PCI and LPC devices. Will stop based on CLKRUN# (and STP_PCI#) in Netbook configurations. |
| Chipset CLK48 | 48.000 MHz | Main Clock Generator | Super I/O, USB controllers. Expected to be shut off during S3 or below in Nettop configurations or S1 or below in Netbook configurations. |
| Chipset CLK14 | 14.31818 MHz | Main Clock Generator | Used for ACPI timer and Multimedia Timers. Expected to be shut off during S3 or below in Nettop configurations or S1 or below in Netbook configurations. |
| LAN_CLK | 5 to 50 MHz | LAN Connect Component | Generated by the LAN Connect component. Expected to be shut off during S3 or below in Nettop configurations or S1 or below in Netbook configurations. |
| SPI_CLK | 17.86 MHz | ICH | Generated by the LAN Connect component. Expected to be shut off during S3 or below in Nettop configurations or S1 or below in Netbook configurations. |

**Figure 4-4. Nettop Only Conceptual System Clock Diagram**

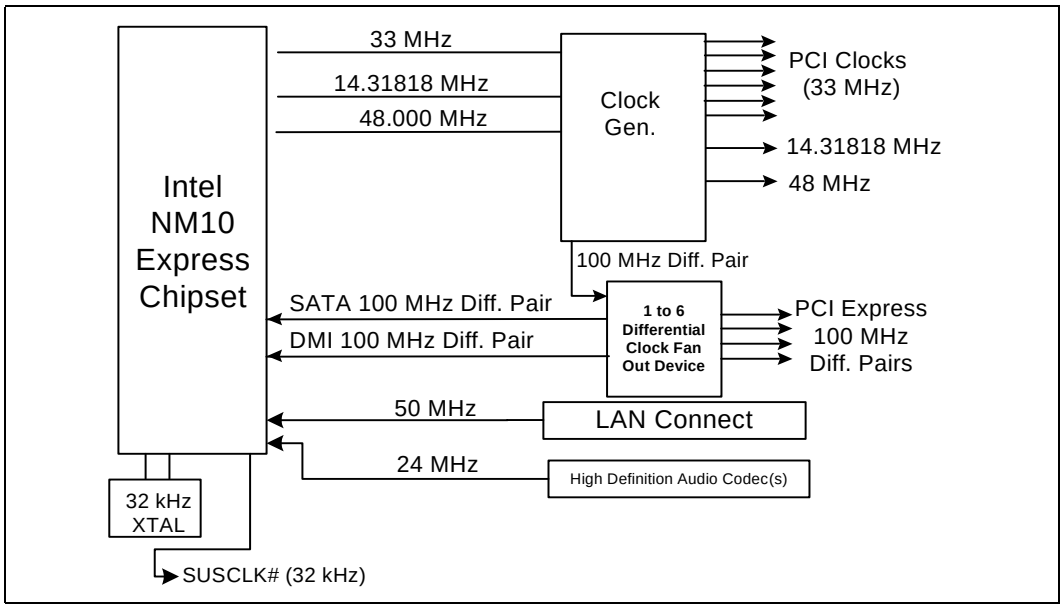
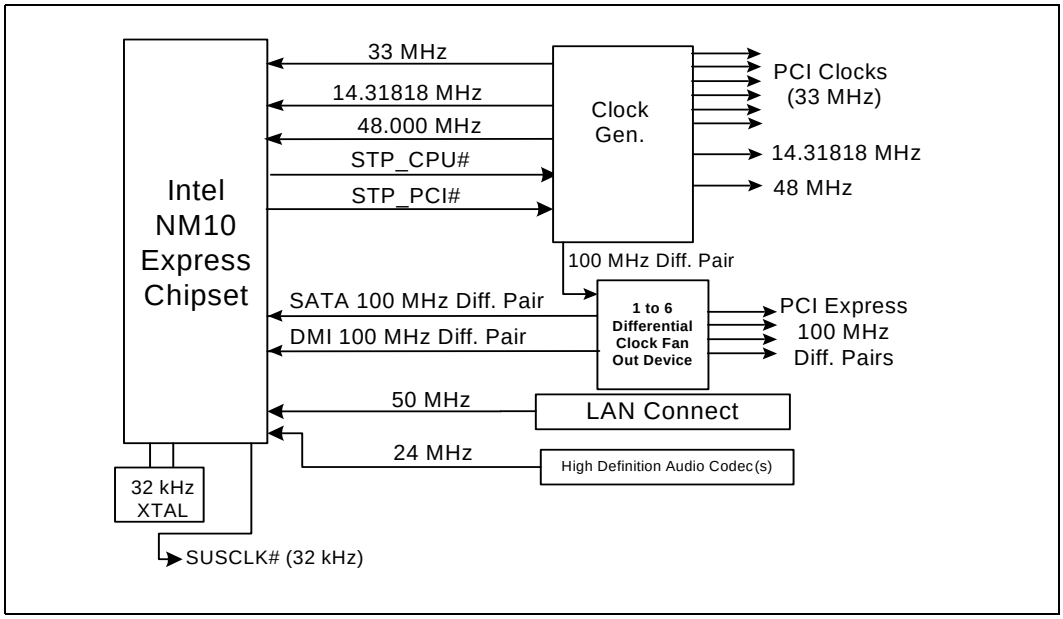
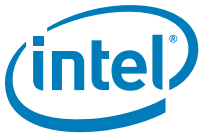



**Figure 4-5. Netbook Only Conceptual Clock Diagram**




§



# 5 Functional Description

This chapter describes the functions and interfaces of Chipset.

## 5.1 PCI-to-PCI Bridge (D30:F0)

The PCI-to-PCI bridge resides in PCI Device 30, Function 0 on bus #0. This portion of Chipset implements the buffering and control logic between PCI and Direct Media Interface (DMI). The arbitration for the PCI bus is handled by this PCI device. The PCI decoder in this device must decode the ranges for the DMI. All register contents are lost when core well power is removed.

Direct Media Interface (DMI) is the chip-to-chip connection between the CPU and I/O Controller Hub. This high-speed interface integrates advanced priority-based servicing allowing for concurrent traffic and true isochronous transfer capabilities. Base functionality is completely software I/O Controller Hub transparent permitting current and legacy software to operate normally.

To provide for true isochronous transfers and configurable Quality of Service (QoS) transactions, Chipset supports two virtual channels on DMI: VC0 and VC1. These two channels provide a fixed arbitration scheme where VC1 is the highest priority. VC0 is the default conduit of traffic for DMI and is always enabled. VC1 must be specifically enabled and configured at both ends of the DMI link (i.e., Chipset and CPU).

Configuration registers for DMI, virtual channel support, and DMI active state power management (ASPM) are in the RCRB space in the Chipset Config Registers (Section 10).

### 5.1.1 PCI Bus Interface

Chipset PCI interface supports *PCI Local Bus Specification,* Revision 2.3, at 33 MHz. Chipset integrates a PCI arbiter that supports up to two external PCI bus masters in addition to the internal Chipset requests.

### 5.1.2 PCI Bridge As an Initiator

The bridge initiates cycles on the PCI bus when granted by the PCI arbiter. The bridge generates the following cycle types:

**Table 5-29.PCI Bridge Initiator Cycle Types**

| Command | C/BE# | Notes |
|---|---|---|
| I/O Read/Write | 2h/3h | Non-posted |

**Table 5-29.PCI Bridge Initiator Cycle Types**

| Command | C/BE# | Notes |
|---|---|---|
| Memory Read/Write | 6h/7h | Writes are posted |
| Configuration Read/Write | Ah/Bh | Non-posted |
| Special Cycles | 1h | Posted |

#### 5.1.2.1 Memory Reads and Writes

The bridge bursts memory writes on PCI that are received as a single packet from DMI.

#### 5.1.2.2 I/O Reads and Writes

The bridge generates single DW I/O read and write cycles. When the cycle completes on the PCI bus, the bridge generates a corresponding completion on DMI. If the cycle is retried, the cycle is kept in the down bound queue and may be passed by a postable cycle.

#### 5.1.2.3 Configuration Reads and Writes

The bridge generates single DW configuration read and write cycles. When the cycle completes on the PCI bus, the bridge generates a corresponding completion. If the cycle is retried, the cycle is kept in the down bound queue and may be passed by a postable cycle.

#### 5.1.2.4 Locked Cycles

The bridge propagates locks from DMI per the *PCI Local Bus Specification*. The PCI bridge implements bus lock, which means the arbiter will not grant to any agent except DMI while locked.

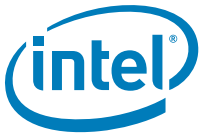
If a locked read results in a target or master abort, the lock is not established (as per the *PCI Local Bus Specification*). Agents north of Chipset must not forward a subsequent locked read to the bridge if they see the first one finish with a failed completion.

#### 5.1.2.5 Target / Master Aborts

When a cycle initiated by the bridge is master/target aborted, the bridge will not re-attempt the same cycle. For multiple DW cycles, the bridge increments the address and attempts the next DW of the transaction. For all non-postable cycles, a target abort response packet is returned for each DW that was master or target aborted on PCI. The bridge drops posted writes that abort.

#### 5.1.2.6 Secondary Master Latency Timer

The bridge implements a Master Latency Timer via the SLT register which, upon expiration, causes the de-assertion of FRAME# at the next legal clock edge when there is another active request to use the PCI bus.



#### 5.1.2.7 Dual Address Cycle (DAC)

The bridge will issue full 64-bit dual address cycles for device memory-mapped registers above 4 GB.

#### 5.1.2.8 Memory and I/O Decode to PCI

The PCI bridge in Chipset is a **subtractive decode agent**, which follows the following rules when forwarding a cycle from DMI to the PCI interface:

- The PCI bridge will **positively** decode any memory/IO address within its window registers, assuming PCICMD.MSE (D30:F0:Offset 04h:bit 1) is set for memory windows and PCICMD.IOSE (D30:F0:Offset 04h:bit 0) is set for IO windows.
- The PCI bridge will **subtractively** decode any 64-bit memory address not claimed by another agent, assuming PCICMD.MSE (D30:F0:Offset 04h:bit 1) is set.
- The PCI bridge will **subtractively** decode any 16-bit I/O address not claimed by another agent assuming PCICMD.IOSE (D30:F0:Offset 04h:bit 0) set
- If BCTRL.IE (D30:F0:Offset 3Eh:bit 2) is set, the PCI bridge will **not positively** forward from primary to secondary called out ranges in the IO window per *PCI Local Bus Specification* (I/O transactions addressing the last 768 bytes in each, 1-KB block: offsets 100h to 3FFh). The PCI bridge will still take them subtractively assuming the above rules.
- If BCTRL.VGAE (D30:F0:Offset 3Eh:bit 3) is set, the PCI bridge will **positively** forward from primary to secondary I/O and memory ranges as called out in the *PCI Bridge Specificatio*n, assuming the above rules are met.

### 5.1.3 Parity Error Detection and Generation

PCI parity errors can be detected and reported. The following behavioral rules apply:

- When a parity error is detected on PCI, the bridge sets the SECSTS.DPE (D30:F0:Offset 1Eh:bit 15).
- If the bridge is a master and BCTRL.PERE (D30:F0:Offset 3Eh:bit 0) and one of the parity errors defined below is detected on PCI, then the bridge will set SECSTS.DPD (D30:F0:Offset 1Eh:bit 8) and will also generate an internal SERR#.
  - — During a write cycle, the PERR# signal is active, or
  - — A data parity error is detected while performing a read cycle
- If an address or command parity error is detected on PCI and PCICMD.SEE (D30:F0:Offset 04h:bit 8), BCTRL.PERE, and BCTRL.SEE (D30:F0:Offset 3Eh:bit 1) are all set, the bridge will set the PSTS.SSE (D30:F0:Offset 06h:bit 14) and generate an internal SERR#.
- If the PSTS.SSE is set because of an address parity error and the PCICMD.SEE is set, the bridge will generate an internal SERR#
- When bad parity is detected from DMI, bad parity will be driven on all data the bridge.

- When an address parity error is detected on PCI, the PCI bridge will not claim the cycle. This is a slight deviation from the PCI bridge spec, which says that a cycle should be claimed if BCTRL.PERE is not set. However, DMI does not have a concept of address parity error, so claiming the cycle could result in the rest of the system seeing a bad transaction as a good transaction.

### 5.1.4 PCIRST#

The PCIRST# pin is generated under two conditions:

- PLTRST# active
- BCTRL.SBR (D30:F0:Offset 3Eh:bit 6) set to 1

The PCIRST# pin is in the resume well. PCIRST# should be tied to PCI bus agents, but not other agents in the system.

### 5.1.5 Peer Cycles

The PCI bridge may be the initiator of peer cycles. Peer cycles include memory, IO, and configuration cycle types. Peer cycles are only allowed through VC0, and are enabled with the following bits:

- BPC.PDE (D30:F0:Offset 4Ch:bit 2) – Memory and IO cycles
- BPC.CDE (D30:F0:Offset 4Ch:bit 1) – Configuration cycles

When enabled for peer for one of the above cycle types, the PCI bridge will perform a peer decode to see if a peer agent can receive the cycle. When not enabled, memory cycles (posted and/or non-posted) are sent to DMI, and I/O and/or configuration cycles are not claimed.

Configuration cycles have special considerations. Under the *PCI Local Bus Specification*, these cycles are not allowed to be forwarded upstream through a bridge. However, to enable things such as manageability, BPC.CDE can be set. When set, type 1 cycles are allowed into the part. The address format of the type 1 cycle is slightly different from a standard PCI configuration cycle to allow addressing of extended PCI space. The format is as follows:

**Table 5-30.Type 1 Address Format**

| Bits | Definition |
|---|---|
| 31:27 | **Reserved** (same as the *PCI Local Bus Specification*) |
| 26:24 | Extended Configuration Address – allows addressing of up to 4K. These bits are combined with bits 7:2 to get the full register. |
| 23:16 | Bus Number (same as the *PCI Local Bus Specification*) |
| 15:11 | Device Number (same as the *PCI Local Bus Specification*) |
| 10:8 | Function Number (same as the *PCI Local Bus Specification*) |

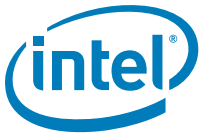


**Table 5-30.Type 1 Address Format**

| Bits | Definition |
|---|---|
| 7:2 | Register (same as the *PCI Local Bus Specification*) |
| 1 | 0 |
| 0 | Must be 1 to indicate a type 1 cycle. Type 0 cycles are not decoded. |

***Note:*** Chipset USB controllers cannot perform peer-to-peer traffic.

## 5.1.6 PCI-to-PCI Bridge Model

From a software perspective, chipset contains a PCI-to-PCI bridge. This bridge connects DMI to the PCI bus. By using the PCI-to-PCI bridge software model, Chipset can have its decode ranges programmed by existing plug-and-play software such that PCI ranges do not conflict with graphics aperture ranges in the Host controller.

***Note:*** All downstream devices should be disabled before reconfiguring the PCI Bridge. Failure to do so may cause undefined results.

## 5.1.7 IDSEL to Device Number Mapping

When addressing devices on the external PCI bus (with the PCI slots), Chipset asserts one address signal as an IDSEL. When accessing device 0, Chipset asserts AD16. When accessing Device 1, Chipset asserts AD17. This mapping continues all the way up to device 15 where Chipset asserts AD31. Note that Chipset internal functions (Intel HD Audio, USB, SATA and PCI Bridge) are enumerated like they are off of a separate PCI bus (DMI) from the external PCI bus. The integrated LAN controller is Device 8 on chipset's PCI bus, and hence it uses AD24 for IDSEL.

## 5.1.8 Standard PCI Bus Configuration Mechanism

The PCI Bus defines a slot based "configuration space" that allows each device to contain up to eight functions with each function containing up to 256, 8-bit configuration registers. The *PCI Local Bus Specification,* Revision 2.3 defines two bus cycles to access the PCI configuration space: Configuration Read and Configuration Write. Memory and I/O spaces are supported directly by the processor. Configuration space is supported by a mapping mechanism implemented within Chipset. The *PCI Local Bus Specification,* Revision 2.3 defines two mechanisms to access configuration space, Mechanism 1 and Mechanism 2. Chipset only supports Mechanism 1.

***Warning:*** Configuration writes to internal devices, when the devices are disabled, are invalid and may cause undefined results.

## 5.2 PCI Express* Root Ports (D28:F0,F1,F2,F3)

There are four root ports available in Chipset. These all reside in device 28, and take function 0 – 3. Port 1 is function 0, port 2 is function 1, port 3 is function 2, port 4 is function 3.

Optionally, PCI Express ports 1-4 can be configured as a single one x4 port identified as port 1. This is accomplished by placing external pull-up resistors on HDA_SDOUT and HDA_SYNC. When these signals are sampled high on PWROK assertion, this will be registered in the Port Configuration field of the Root Port Configuration Register and the corresponding ports will be configured as one x4 port.

### 5.2.1 Interrupt Generation

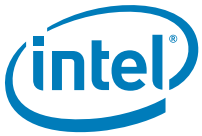
The root port generates interrupts on behalf of Hot-Plug and power management events, when enabled. These interrupts can either be pin based, or can be MSIs, when enabled.

When an interrupt is generated via the legacy pin, the pin is internally routed to Chipset interrupt controllers. The pin that is driven is based upon the setting of the chipset configuration registers. Specifically, the chipset configuration registers used are the D28IP (Base address + 310Ch) and D28IR (Base address + 3146h) registers.

Table 5-31 summarizes interrupt behavior for MSI and wire-modes. In the table “bits” refers to the Hot-Plug and PME interrupt bits.

**Table 5-31.MSI vs. 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 |



## 5.2.2 Power Management

### 5.2.2.1 S3/S4/S5 Support

Software initiates the transition to S3/S4/S5 by performing an IO write to the Power Management Control register in Chipset. After the IO write completion has been returned to the processor, each root port will send a PME_Turn_Off TLP (Transaction Layer Packet) message on it's downstream link. The device attached to the link will eventually respond with a PME_TO_Ack TLP message followed by sending a PM_Enter_L23 DLLP (Data Link Layer Packet) request to enter the L2/L3 Ready state. When all of Chipset root ports links are in the L2/L3 Ready state, Chipset power management control logic will proceed with the entry into S3/S4/S5.

Prior to entering S3, software is required to put each device into $D3_{HOT}$. When a device is put into $D3_{HOT}$, it will initiate entry into a L1 link state by sending a PM_Enter_L1 DLLP. Thus, 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 Chipset can make no assumptions about the state of the link prior to receiving a PME_Turn_Off message.

### 5.2.2.2 Resuming from Suspended State

The root port contains enough circuitry in the resume well to detect a wake event thru the WAKE# signal and to wake the system. When WAKE# is detected asserted, an internal signal is sent to the power management controller of Chipset to cause the system to wake up. This internal message is not logged in any register, nor is an interrupt/GPE generated due to it.

### 5.2.2.3 Device Initiated PM_PME Message

When the system has returned to a working state from a previous low power state, a device requesting service will send a PM_PME message continuously, until acknowledge by the root port. The root port will take different actions depending upon whether this is the first PM_PME 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 - D28:F0/F1/F2/F3:Offset 60h:bit 16 is cleared), the root port will set RSTS.PS, and log the PME Requester ID into RSTS.RID (D28:F0/F1/F2/F3:Offset 60h:bits 15:0). If an interrupt is enabled via RCTL.PIE (D28:F0/F1/F2/F3:Offset 5Ch:bit 3), an interrupt will be generated. This interrupt can be either a pin or a MSI if MSI is enabled via MC.MSIE (D28:F0/F1/F2/F3:Offset 82h:bit 0). See Section 5.2.2.4 for SMI/SCI generation.

If this is a subsequent message received (RSTS.PS is already set), the root port will set RSTS.PP (D28:F0/F1/F2/F3:Offset 60h:bit 17) and log the PME Requester ID from the message in a hidden register. 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 from the hidden register into RSTS.RID.

If RCTL.PIE is set, generate an interrupt. If RCTL.PIE is not set, send over 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 0 to a 1, and interrupt must 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.

#### 5.2.2.4 SMI/SCI Generation

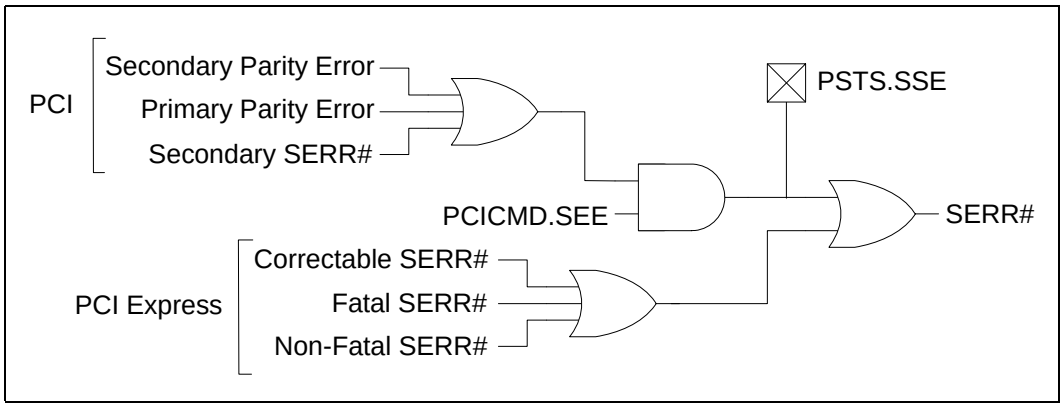
Interrupts for power management events are not supported on legacy operating systems. To support power management on non-PCI Express aware operating systems, PM events can be routed to generate SCI. To generate SCI, MPC.PMCE must be set. When set, a power management event will cause SMSCS.PMCS (D28:F0/F1/F2/F3:Offset DCh:bit 31) to be set.

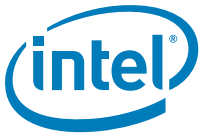
Additionally, BIOS workarounds for power management can be supported by setting MPC.PMME (D28:F0/F1/F2/F3:Offset D8h:bit 0). When this bit is set, power management events will set SMSCS.PMMS (D28:F0/F1/F2/F3:Offset DCh:bit 0), and SMI # will be generated. This bit will be set regardless of whether interrupts or SCI is enabled. The SMI# may occur concurrently with an interrupt or SCI.

### 5.2.3 SERR# Generation

SERR# may be generated via 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 5-6. Generation of SERR# to Platform**





### 5.2.4 Hot-Plug

Each root port implements a Hot-Plug controller which performs the following:

- Messages to turn on / off / blink LEDs
- Presence and attention button detection
- Interrupt generation

The root port only allows Hot-Plug with modules (e.g., ExpressCard*). Edge-connector based Hot-Plug is not supported.

#### 5.2.4.1 Presence Detection

When a module is plugged in and power is supplied, the physical layer will detect the presence of the device, and the root port sets SLSTS.PDS (D28:F0/F1/F2/F3:Offset 5Ah:bit 6) and SLSTS.PDC (D28:F0/F1/F2/F3:Offset 6h:bit 3). If SLCTL.PDE (D28:F0/F1/F2/F3:Offset 58h:bit 3) and SLCTL.HPE (D28:F0/F1/F2/F3:Offset 58h:bit 5) are both set, the root port will also generate an interrupt.

When a module is removed (via 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.

#### 5.2.4.2 Message Generation

When system software writes to SLCTL.AIC (D28:F0/F1/F2/F3:Offset 58h:bits 7:6) or SLCTL.PIC (D28:F0/F1/F2/F3:Offset 58h:bits 9:8), the root port will send a message down the link to change the state of LEDs on the module.

Writes to these fields are non-postable cycles, and the resulting message is a postable cycle. When receiving one of these writes, the root port performs the following:

- Changes the state in the register.
- Generates a completion into the upstream queue
- Formulates a message for the downstream port if the field is written to regardless of if the field changed.
- Generates the message on the downstream port
- When the last message of a command is transmitted, sets SLSTS.CCE (D28:F0/F1/F2/F3:Offset 58h:bit 4) to indicate the command has completed. If SLCTL.CCE and SLCTL.HPE (D28:F0/F1/F2/F3:Offset 58h:bit 5) are set, the root port generates an interrupt.

The command completed register (SLSTS.CC) applies only to commands issued by software to control the Attention Indicator (SLCTL.AIC), Power Indicator (SLCTL.PIC), or Power Controller (SLCTL.PCC). However, writes to other parts of the Slot Control Register would invariably end up writing to the indicators, power controller fields; Hence, any write to the Slot Control Register is considered a command and if enabled,

will result in a command complete interrupt. The only exception to this rule is a write to disable the command complete interrupt which will not result in a command complete interrupt.

A single write to the Slot Control register is considered to be a single command, and hence receives a single command complete, even if the write affects more than one field in the Slot Control Register.

#### 5.2.4.3 Attention Button Detection

When an attached device is ejected, an attention button could be pressed by the user. This attention button press will result in a the PCI Express message "Attention_Button_Pressed" from the device. Upon receiving this message, the root port will set SLSTS.ABP (D28:F0/F1/F2/F3:Offset 5Ah:bit 0).

If SLCTL.ABE (D28:F0/F1/F2/F3:Offset 58h:bit 0) and SLCTL.HPE (D28:F0/F1/F2/F3:Offset 58h:bit 5) are set, the Hot-Plug controller will also generate an interrupt. The interrupt is generated on an edge-event. For example, if SLSTS.ABP is already set, a new interrupt will not be generated.

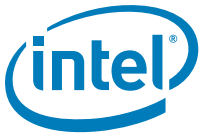
#### 5.2.4.4 SMI/SCI Generation

Interrupts for Hot-Plug events are not supported on legacy operating systems. To support Hot-Plug on non-PCI Express aware operating systems, Hot-Plug events can be routed to generate SCI. To generate SCI, MPC.HPCE (D28:F0/F1/F2/F3:Offset D8h:bit 30) must be set. When set, enabled Hot-Plug events will cause SMSCS.HPCS (D28:F0/F1/F2/F3:Offset DCh:bit 30) to be set.

Additionally, BIOS workarounds for Hot-Plug can be supported by setting MPC.HPME (D28:F0/F1/F2/F3:Offset D8h:bit 1). 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:

- Command Completed - SCSCS.HPCCM (D28:F0/F1/F2/F3:Offset DCh:bit 3)
- Presence Detect Changed - SMSCS.HPPDM (D28:F0/F1/F2/F3:Offset DCh:bit 1)
- Attention Button Pressed - SMSCS.HPABM (D28:F0/F1/F2/F3:Offset DCh:bit 2)
- Link Active State Changed - SMSCS.HPLAS (D28:F0/F1/F2/F3:Offset DCh:bit 4)

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.



## 5.3 LAN Controller (B1:D8:F0)

Chipset's integrated LAN controller includes a 32-bit PCI controller that provides enhanced scatter-gather bus mastering capabilities and enables the LAN controller to perform high-speed data transfers over the PCI bus. Its bus master capabilities enable the component to process high level commands and perform multiple operations; this lowers processor utilization by off-loading communication tasks from the processor. Two large transmit and receive FIFOs of 3 KB each, help prevent data underruns and overruns while waiting for bus accesses. This enables the integrated LAN controller to transmit data with minimum interframe spacing (IFS).

Chipset integrated LAN controller can operate in either full-duplex or half-duplex mode. In full- duplex mode the LAN controller adheres with the *IEEE 802.3x Flow Control Specification*. Half duplex performance is enhanced by a proprietary collision reduction mechanism.

The integrated LAN controller also includes an interface to a serial (4-pin) EEPROM. The EEPROM provides power-on initialization for hardware and software configuration parameters.

From a software perspective, the integrated LAN controller appears to reside on the secondary side of chipset's virtual PCI-to-PCI bridge (see Section 5.1.6). This is typically Bus 1, but may be assigned a different number, depending upon system configuration.

The following summarizes Chipset LAN controller features:

- Compliance with Advanced Configuration and Power Interface and PCI Power Management standards
- Support for wake-up on interesting packets and link status change
- Support for remote power-up using Wake on LAN* (WOL) technology
- Deep power-down mode support
- Backward compatible software with 82550, 82557, 82558 and 82559
- TCP/UDP checksum off load capabilities
- Support for Intel's Adaptive Technology

### 5.3.1 LAN Controller PCI Bus Interface

As a Fast Ethernet controller, the role of Chipset integrated LAN controller is to access transmitted data or deposit received data. The LAN controller, as a bus master device, initiates memory cycles via the PCI bus to fetch or deposit the required data.

To perform these actions, the LAN controller is controlled and examined by the processor via its control and status structures and registers. Some of these control and status structures reside in the LAN controller and some reside in system memory. For access to the LAN controller's Control/Status Registers (CSR), the LAN controller acts as a slave (in other words, a target device). The LAN controller serves as a slave also while the processor accesses the EEPROM.

#### 5.3.1.1 Bus Slave Operation

Chipset integrated LAN controller serves as a target device in one of the following cases:

- Processor accesses to the LAN controller System Control Block (SCB) Control/ Status Registers (CSR)
- Processor accesses to the EEPROM through its CSR
- Processor accesses to the LAN controller PORT address via the CSR
- Processor accesses to the MDI control register in the CSR

The size of the CSR memory space is 4 Kbyte in the memory space and 64 bytes in the I/O space. The LAN controller treats accesses to these memory spaces differently.

**Control/Status Register (CSR) Accesses**

The integrated LAN controller supports zero wait-state single cycle memory or I/O mapped accesses to its CSR space. Separate BARs request 4 KB of memory space and 64 bytes of I/O space to accomplish this. Based on its needs, the software driver uses either memory or I/O mapping to access these registers. The LAN controller provides four valid KB of CSR space that include the following elements:

- System Control Block (SCB) registers
- PORT register
- EEPROM control register
- MDI control register
- Flow control registers

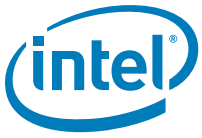
In the case of accessing the Control/Status Registers, the processor is the initiator and the LAN controller is the target.

**Retry Premature Accesses**

The LAN controller responds with a Retry to any configuration cycle accessing the LAN controller before the completion of the automatic read of the EEPROM. The LAN controller may continue to Retry any configuration accesses until the EEPROM read is complete. The LAN controller does not enforce the rule that the retried master must attempt to access the same address again in order to complete any delayed transaction. Any master access to the LAN controller after the completion of the EEPROM read is honored.

**Error Handling**

**Data Parity Errors:** The LAN controller checks for data parity errors while it is the target of the transaction. If an error was detected, the LAN controller sets the Detected Parity Error bit in the PCI Configuration Status register, bit 15. The LAN controller also



asserts PERR#, if the Parity Error Response bit is set (PCI Configuration Command register, bit 6). The LAN controller does not attempt to terminate a cycle in which a parity error was detected. This gives the initiator the option of recovery.

**Target-Disconnect:** The LAN controller prematurely terminate a cycle in the following cases:

- After accesses to its CSR
- After accesses to the configuration space

**System Error:** The LAN controller reports parity error during the address phase using the SERR# pin. If the SERR# Enable bit in the PCI Configuration Command register or the Parity Error Response bit are not set, the LAN controller only sets the Detected Parity Error bit (PCI Configuration Status register, bit 15). If SERR# Enable and Parity Error Response bits are both set, the LAN controller sets the Signaled System Error bit (PCI Configuration Status register, bit 14) as well as the Detected Parity Error bit and asserts SERR# for one clock.

The LAN controller, when detecting system error, claims the cycle if it was the target of the transaction and continues the transaction as if the address was correct.

***Note:*** The LAN controller reports a system error for any error during an address phase, whether or not it is involved in the current transaction.

#### 5.3.1.2 CLKRUN# Signal (Netbook Only)

Chipset receives a free-running 33 MHz clock. It does not stop based on the CLKRUN# signal and protocol. When the LAN controller runs cycles on the PCI bus, Chipset makes sure that the STP_PCI# signal is high indicating that the PCI clock will be running. This is to make sure that any PCI tracker does not get confused by transactions on the PCI bus with its PCI clock stopped.

#### 5.3.1.3 PCI Power Management

Enhanced support for the power management standard, *PCI Local Bus Specification,* Revision 2.3, is provided in Chipset integrated LAN controller. The LAN controller supports a large set of wake-up packets and the capability to wake the system from a low power state on a link status change. The LAN controller enables the host system to be in a sleep state and remain virtually connected to the network.

After a power management event or link status change is detected, the LAN controller wakes the host system. The following sections describe these events, the LAN controller power states, and estimated power consumption at each power state.

The LAN controller contains power management registers for PCI, and implements four power states, D0 through D3, which vary from maximum power consumption at D0 to the minimum power consumption at D3. PCI transactions are only allowed in the D0 state, except for host accesses to the LAN controller's PCI configuration registers. The D1 and D2 power management states enable intermediate power savings while

providing the system wake-up capabilities. In the $D3_{COLD}$ state, the LAN controller can provide wake-up capabilities. Wake-up indications from the LAN controller are provided by the Power Management Event (PME#) signal.

#### 5.3.1.4 PCI Reset Signal

The PCIRST# signal may be activated in one of the following cases:

- During S3–S5 states
- Due to a CF9h reset

If PME is enabled (in the PCI power management registers), PCIRST# assertion does not affect any PME related circuits (in other words, PCI power management registers and the wake-up packet would not be affected). While PCIRST# is active, the LAN controller ignores other PCI signals. The configuration of the LAN controller registers associated with ACPI wake events is not affected by PCIRST#.

The integrated LAN controller uses the PCIRST# or the PWROK signal as an indication to ignore the PCI interface. Following the deassertion of PCIRST#, the LAN controller PCI Configuration Space, MAC configuration, and memory structure are initialized while preserving the PME# signal and its context.

#### 5.3.1.5 Wake-Up Events

There are two types of wake-up events: “Interesting” Packets and Link Status Change. These two events are detailed below.

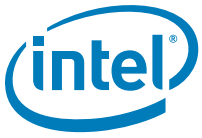
***Note:*** If the Wake on LAN bit in the EEPROM is not set, wake-up events are supported only if the PME Enable bit in the Power Management Control/Status Register (PMCSR) is set. However, if the Wake on LAN bit in the EEPROM is set, and Wake on Magic Packet* or Wake on Link Status Change are enabled, the Power Management Enable bit is ignored with respect to these events. In the latter case, PME# would be asserted by these events.

**“Interesting” Packet Event**

In the power-down state, the LAN controller is capable of recognizing “interesting” packets. The LAN controller supports predefined and programmable packets that can be defined as any of the following:

- ARP Packets (with Multiple IP addresses)
- Direct Packets (with or without type qualification)
- Magic Packet
- Neighbor Discovery Multicast Address Packet (‘ARP’ in IPv6 environment)
- NetBIOS over TCP/IP (NBT) Query Packet (under IPv4)
- Internetwork Package Exchange* (IPX) Diagnostic Packet

This allows the LAN controller to handle various packet types. In general, the LAN controller supports programmable filtering of any packet in the first 128 bytes.



When the LAN controller is in one of the low power states, it searches for a predefined pattern in the first 128 bytes of the incoming packets. The only exception is the Magic Packet, which is scanned for the entire frame. The LAN controller classifies the incoming packets as one of the following categories:

- **No Match:** The LAN controller discards the packet and continues to process the incoming packets.
- **TCO Packet:** The LAN controller implements perfect filtering of TCO packets. After a TCO packet is processed, the LAN controller is ready for the next incoming packet. TCO packets are treated as any other wake-up packet and may assert the PME# signal if configured to do so.
- **Wake-up Packet:** The LAN controller is capable of recognizing and storing the first 128 bytes of a wake-up packet. If a wake-up packet is larger than 128 bytes, its tail is discarded by the LAN controller. After the system is fully powered-up, software has the ability to determine the cause of the wake-up event via the PMDR and dump the stored data to the host memory.

  Magic Packets are an exception. The Magic Packets may cause a power management event and set an indication bit in the PMDR; however, it is not stored by the LAN controller for use by the system when it is woken up.

#### Link Status Change Event

The LAN controller link status indication circuit is capable of issuing a PME on a link status change from a valid link to an invalid link condition or vice versa. The LAN controller reports a PME link status event in all power states. If the Wake on LAN bit in the EEPROM is not set, the PME# signal is gated by the PME Enable bit in the PMCSR and the CSMA Configure command.

### 5.3.1.6 Wake on LAN* (Preboot Wake-Up)

The LAN controller enters Wake on LAN mode after reset if the Wake on LAN bit in the EEPROM is set. At this point, the LAN controller is in the D0u state. When the LAN controller is in Wake on LAN mode:

- The LAN controller scans incoming packets for a Magic Packet and asserts the PME# signal for 52 ms when a 1 is detected in Wake on LAN mode.
- The Activity LED changes its functionality to indicates that the received frame passed Individual Address (IA) filtering or broadcast filtering.
- The PCI Configuration registers are accessible to the host.

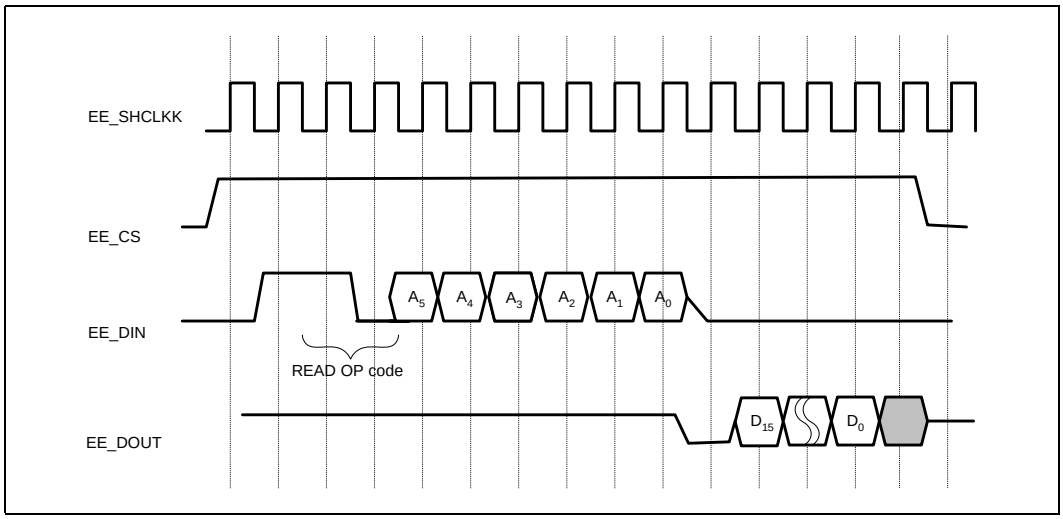
The LAN controller switches from Wake on LAN mode to the D0a power state following a setup of the Memory or I/O Base Address Registers in the PCI configuration space.

### 5.3.2 Serial EEPROM Interface

The serial EEPROM stores configuration data for Chipset integrated LAN controller and is a serial in/serial out device. The LAN controller supports a 64-register or 256-register size EEPROM and automatically detects the EEPROM's size. The EEPROM should operate at a frequency of at least 1 MHz.

All accesses, either read or write, are preceded by a command instruction to the device. The address field is six bits for a 64-register EEPROM or eight bits for a 256-register EEPROM. The end of the address field is indicated by a dummy 0 bit from the EEPROM that indicates the entire address field has been transferred to the device. An EEPROM read instruction waveform is shown in Figure 5-7.

**Figure 5-7. 64-Word EEPROM Read Instruction Waveform**



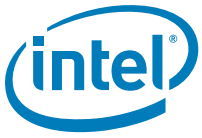

The LAN controller performs an automatic read of seven words (0h, 1h, 2h, Ah, Bh, Ch, and Dh) of the EEPROM after the deassertion of Reset.

### 5.3.3 CSMA/CD Unit

Chipset integrated LAN controller CSMA/CD unit implements both the IEEE 802.3 Ethernet 10 Mbps and IEEE 802.3u Fast Ethernet 100 Mbps standards. It performs all the CSMA/CD protocol functions (e.g., transmission, reception, collision handling, etc.). The LAN controller CSMA/CD unit interfaces to the 82562ET/EM/EZ/EX 10/100 Mbps Ethernet through chipset's LAN Connect interface signals.

#### 5.3.3.1 Full Duplex

When operating in full-duplex mode, the LAN controller can transmit and receive frames simultaneously. Transmission starts regardless of the state of the internal receive path. Reception starts when the platform LAN Connect component detects a valid frame on its receive differential pair. Chipset integrated LAN controller also supports the IEEE 802.3x flow control standard, when in full-duplex mode.



The LAN controller operates in either half-duplex mode or full-duplex mode. For proper operation, both the LAN controller CSMA/CD module and the discrete platform LAN Connect component must be set to the same duplex mode. The CSMA duplex mode is set by the LAN Controller Configure command or forced by automatically tracking the mode in the platform LAN Connect component. Following reset, the CSMA defaults to automatically track the platform LAN Connect component duplex mode.

The selection of duplex operation (full or half) and flow control is done in two levels: MAC and LAN Connect.

#### 5.3.3.2 Flow Control

The LAN controller supports IEEE 802.3x frame-based flow control frames only in both full duplex and half duplex switched environments. The LAN controller flow control feature is not intended to be used in shared media environments.

Flow control is optional in full-duplex mode and is selected through software configuration. There are three modes of flow control that can be selected: frame-based transmit flow control, frame-based receive flow control, and none.

#### 5.3.3.3 VLAN Support

The LAN controller supports the IEEE 802.1 standard VLAN. All VLAN flows will be implemented by software. The LAN controller supports the reception of long frames, specifically frames longer than 1518 bytes, including the CRC, if software sets the Long Receive OK bit in the Configuration command. Otherwise, “long” frames are discarded.

### 5.3.4 Media Management Interface

The management interface allows the processor to control the platform LAN Connect component via a control register in Chipset integrated LAN controller. This allows the software driver to place the platform LAN Connect in specific modes (e.g., full duplex, loopback, power down, etc.) without the need for specific hardware pins to select the desired mode. This structure allows the LAN controller to query the platform LAN Connect component for status of the link. This register is the MDI Control Register and resides at offset 10h in the LAN controller CSR. The MDI registers reside within the platform LAN Connect component, and are described in detail in the platform LAN Connect component’s datasheet. The processor writes commands to this register and the LAN controller reads or writes the control/status parameters to the platform LAN Connect component through the MDI register.

### 5.3.5 TCO Functionality

Chipset integrated LAN controller supports management communication to reduce Total Cost of Ownership (TCO). The SMBus is used as an interface between the ASF controller and the integrated TCO host controller. There are two different types of TCO operation that are supported (only one supported at a time), they are 1) Integrated ASF Control or 2) external TCO controller support. The SMLink is a dedicated bus

between the LAN controller and the integrated ASF controller (if enabled) or an external management controller. An EEPROM of 256 words is required to support the heartbeat command.

#### 5.3.5.1 Advanced TCO Mode

The Advanced TCO functionalities through the SMLink are listed in Table 5-32.

**Table 5-32.Advanced TCO Functionality**

| Power State | TCO Controller Functionality |
|---|---|
| D0 nominal | Transmit<br>Set Receive TCO Packets<br>Receive TCO Packets<br>Read Chipset status (PM & Link state)<br>Force TCO Mode |
| Dx (x>0) | D0 functionality plus:<br>Read PHY registers |
| Force TCO Mode | Dx functionality plus:<br>Configuration commands<br>Read/Write PHY registers |

***Note:*** For a complete description on various commands, see the *Total Cost of Ownership (TCO) System Management Bus Interface Application Note (AP-430)*.

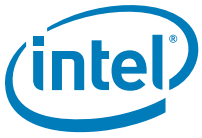
**Transmit Command during Normal Operation**

To serve a transmit request from the TCO controller, Chipset LAN controller first completes the current transmit DMA, sets the TCO request bit in the PMDR register (see Section 11.2), and then responds to the TCO controller's transmit request. Following the completion of the TCO transmit DMA, the LAN controller increments the Transmit TCO statistic counter (described in Section 11.2.14). Following the completion of the transmit operation, Chipset increments the nominal transmit statistic counters, clears the TCO request bit in the PMDR register, and resumes its normal transmit flow. The receive flow is not affected during this entire period of time.

**Receive TCO**

Chipset LAN controller supports receive flow towards the TCO controller. Chipset can transfer only TCO packets, or all packets that passed MAC address filtering according to its configuration and mode of operation as detailed below. While configured to transfer only TCO packets, it supports Ethernet type II packets with optional VLAN tagging.

**Force TCO Mode:** While Chipset is in the force TCO mode, it may receive packets (TCO or all) directly from the TCO controller. Receiving TCO packets and filtering level is controlled by the set Receive enable command from the TCO controller. Following a reception of a TCO packet, Chipset increments its nominal Receive statistic counters as well as the Receive TCO counter.



**Dx>0 Power State:** While Chipset is in a powerdown state, it may receive TCO packets or all directly to the TCO controller. Receiving TCO packets is enabled by the set Receive enable command from the TCO controller. Although TCO packet might match one of the other wake up filters, once it is transferred to the TCO controller, no further matching is searched for and PME is not issued. While receive to TCO is not enabled, a TCO packet may cause a PME if configured to do so (setting TCO to 1 in the filter type).

**D0 Power State:** At D0 power state, Chipset may transfer TCO packets to the TCO controller. At this state, TCO packets are posted first to the host memory, then read by Chipset, and then posted back to the TCO controller. After the packet is posted to TCO, the receive memory structure (that is occupied by the TCO packet) is reclaimed. Other than providing the necessary receive resources, there is no required device driver intervention with this process. Eventually, Chipset increments the receive TCO static counter, clears the TCO request bit, and resumes normal control.

#### Read Chipset Status (PM and Link State)

The TCO controller is capable of reading Chipset power state and link status. Following a status change, Chipset asserts LINKALERT# and then the TCO can read its new power state.

#### Set Force TCO Mode

The TCO controller put Chipset into the Force TCO mode. Chipset is set back to the nominal operation following a PCIRST#. Following the transition from nominal mode to a TCO mode, Chipset aborts transmission and reception and loses its memory structures. The TCO may configure Chipset before it starts transmission and reception if required.

***Warning:*** The Force TCO is a destructive command. It causes Chipset to lose its memory structures, and during the Force TCO mode Chipset ignores any PCI accesses. Therefore, it is highly recommended to use this command by the TCO controller at system emergency only.

## 5.4 Alert Standard Format (ASF)

The ASF controller collects information from various components in the system (including the processor, chipset, BIOS, and sensors on the motherboard) and sends this information via the LAN controller to a remote server running a management console. The controller also accepts commands back from the management console and drives the execution of those commands on the local system.

The ASF controller is responsible for monitoring sensor devices and sending packets through the LAN controller SMBus (System Management Bus) interface. These ASF controller alerting capabilities include system health information such as BIOS messages, POST alerts, operating system failure notifications, and heartbeat signals to indicate the system is accessible to the server. Also included are environmental notification (e.g., thermal, voltage and fan alerts) that send proactive warnings that something is wrong with the hardware. The packets are used as Alert (S.O.S.) packets

or as “heartbeat” status packets. In addition, asset security is provided by messages (e.g., “cover tamper” and “CPU missing”) that notify of potential system break-ins and processor or memory theft.

The ASF controller is also responsible for receiving and responding to RMCP (Remote Management and Control Protocol) packets. RMCP packets are used to perform various system APM commands (e.g., reset, power-up, power-cycle, and power-down). RMCP can also be used to ping the system to ensure that it is on the network and running correctly and for capability reporting. A major advantage of ASF is that it provides these services during the time that software is unable to do so (e.g., during a low-power state, during boot-up, or during an operating system hang) but are not precluded from running in the working state.

The ASF controller communicates to the system and the LAN controller logic through the SMBus connections. The first SMBus connects to the host SMBus controller (within Chipset) and any SMBus platform sensors. The SMBus host is accessible by the system software, including software running on the operating system and the BIOS. Note that the host side bus may require isolation if there are non-auxiliary devices that can pull down the bus when un-powered. The second SMBus connects to the LAN controller. This second SMBus is used to provide a transmit/receive network interface.

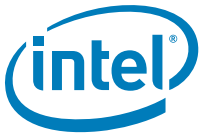
The stimulus for causing the ASF controller to send packets can be either internal or external to the ASF controller. External stimuli are link status changes or polling data from SMBus sensor devices; internal events come from, among others, a set of timers or an event caused by software.

The ASF controller provides three local configuration protocols via the host SMBus. The first one is the SMBus ARP interface that is used to identify the SMBus device and allow dynamic SMBus address assignment. The second protocol is the ASF controller command set that allows software to manage an ASF controller compliant interface for retrieving info, sending alerts, and controlling timers.

Chipset provides an input and an output EEPROM interface. The EEPROM contains the LAN controller configuration and the ASF controller configuration/packet information.

### 5.4.1 ASF Management Solution Features/Capabilities

- Alerting
  - — Transmit SOS packets from S0–S5 states
  - — System Health Heartbeats
  - — SOS Hardware Events
    - System Boot Failure (Watchdog Expires on boot)
    - LAN Link Loss
    - Entity Presence (on ASF power-up)
    - SMBus Hung
    - Maximum of eight Legacy Sensors
    - Maximum of 128 ASF Sensor events
  - — Watchdog Timer for operating system lockup/System Hang/Failure to Boot



  - General Push support for BIOS (POST messages)
- Remote Control
  - Presence Ping Response
  - Configurable Boot Options
  - Capabilities Reporting
  - Auto-ARP Support
  - System Remote Control
    - Power-Down
    - Power-Up
    - Power Cycle
    - System Reset
  - State-Based Security – Conditional Action on WatchDog Expire
- ASF Compliance
  - Compliant with the *Alert Standard Format (ASF) Specification,* Version 1.03
    - PET Compliant Packets
    - RMCP
    - Legacy Sensor Polling
    - ASF Sensor Polling
    - Remote Control Sensor Support
- Advanced Features / Miscellaneous
  - SMBus 2.0 compliant
  - Optional reset extension logic (for use with a power-on reset)

### 5.4.2 ASF Hardware Support

ASF requires additional hardware to make a complete solution.

***Note:*** If an ASF compatible device is externally connected and properly configured, the internal Chipset ASF controller will be disabled. The external ASF device will have access to the SMBus controller.

#### 5.4.2.1 Intel® 82562EM/EX

The 82562EM/EX Ethernet LAN controller is necessary. This LAN controller provides the means of transmitting and receiving data on the network, as well as adding the Ethernet CRC to the data from the ASF.

#### 5.4.2.2 EEPROM (256x16, 1 MHz)

To support Chipset ASF solution, a larger, 256x16 1 MHz, EEPROM is necessary to configure defaults on reset and on hard power losses (software un-initiated). The ASF controller shares this EEPROM with the LAN controller and provides a pass through interface to achieve this. The ASF controller expects to have exclusive access to words

40h through F7h. The LAN controller can use the other EEPROM words. The ASF controller will default to safe defaults if the EEPROM is not present or not configured properly (both cause an invalid CRC).

#### 5.4.2.3 Legacy Sensor SMBus Devices

The ASF controller is capable of monitoring up to eight sensor devices on the main SMBus. These sensors are expected to be compliant with the Legacy Sensor Characteristics defined in the *Alert Standard Format (ASF) Specification,* Version 1.03.

#### 5.4.2.4 Remote Control SMBus Devices

The ASF controller is capable of causing remote control actions to Remote Control devices via SMBus. These remote control actions include Power-Up, Power-Down, Power-Cycle, and Reset. The ASF controller supports devices that conform to the *Alert Standard Format (ASF) Specification,* Version 1.03, Remote Control Devices.

#### 5.4.2.5 ASF Sensor SMBus Devices

The ASF controller is capable of monitoring up to 128 ASF sensor devices on the main SMBus. However, ASF is restricted by the number of total events which may reduce the number of SMBus devices supported. The maximum number of events supported by ASF is 128. The ASF sensors are expected to operate as defined in the *Alert Standard Format (ASF) Specification,* Version 1.03.

### 5.4.3 ASF Software Support

ASF requires software support to make a complete solution. The following software is used as part of the complete solution.

- ASF Configuration driver / application
- Network Driver
- BIOS Support for SMBIOS, SMBus ARP, ACPI
- Sensor Configuration driver / application

***Note:*** Contact your Intel Field Representative for the Client ASF Software Development Kit (SDK) that includes additional documentation and a copy of the client ASF software drivers. Intel also provides an ASF Console SDK to add ASF support to a management console.

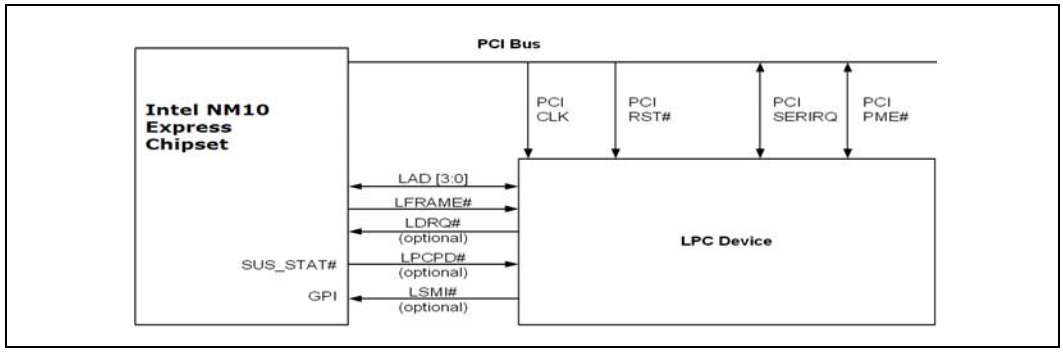
# 5.5 LPC Bridge (w/ System and Management Functions) (D31:F0)

The LPC bridge function of Chipset resides in PCI Device 31:Function 0. In addition to the LPC bridge function, D31:F0 contains other functional units including DMA, Interrupt controllers, Timers, Power Management, System Management, GPIO, and RTC. In this chapter, registers and functions associated with other functional units (power management, GPIO, USB, IDE, etc.) are described in their respective sections.

## 5.5.1 LPC Interface

Chipset implements an LPC interface as described in the *Low Pin Count Interface Specification,* Revision 1.1. The LPC interface to Chipset is shown in Figure 5-8. Note that Chipset implements all of the signals that are shown as optional, but peripherals are not required to do so.

**Figure 5-8. LPC Interface Diagram**




### 5.5.1.1 LPC Cycle Types

Chipset implements the following cycle types as described in Table 5-33

**Table 5-33. LPC Cycle Types Supported**

| Cycle Type | Comment |
|---|---|
| I/O Read | 1 byte only. Chipset breaks up 16- and 32-bit processor cycles into multiple 8-bit transfers. |
| I/O Write | 1 byte only. Chipset breaks up 16- and 32-bit processor cycles into multiple 8-bit transfers. |
| DMA Read | Can be 1, or 2 bytes |
| DMA Write | Can be 1, or 2 bytes |
| Bus Master Read | Can be 1, 2, or 4 bytes. (See Note 1 below) |
| Bus Master Write | Can be 1, 2, or 4 bytes. (See Note 1 below) |

**NOTES:**

1. Bus Master Read or Write cycles must be naturally aligned. For example, a 1-byte transfer can be to any address. However, the 2-byte transfer must be word-aligned (i.e., with an address where A0=0). A dword transfer must be dword-aligned (i.e., with an address where A1 and A0 are both 0)

#### 5.5.1.2 Start Field Definition

**Table 5-34.Start Field Bit Definitions**

| Bits[3:0] Encoding | Definition |
|---|---|
| 0000 | Start of cycle for a generic target |
| 0010 | Grant for bus master 0 |
| 0011 | Grant for bus master 1 |
| 1111 | Stop/Abort: End of a cycle for a target. |

**NOTE:** All other encodings are Reserved.

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

Chipset drives bit 0 of this field to 0. Peripherals running bus master cycles must also drive bit 0 to 0. Table 5-35 shows the valid bit encodings.

**Table 5-35.Cycle Type Bit Definitions**

| Bits[3:2] | Bit1 | Definition |
|---|---|---|
| 00 | 0 | I/O Read |
| 00 | 1 | I/O Write |
| 10 | 0 | DMA Read |
| 10 | 1 | DMA Write |
| 11 | x | **Reserved**. If a peripheral performing a bus master cycle generates this value, Chipset aborts the cycle. |

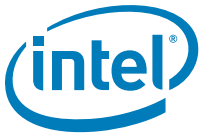
**NOTE:** All other encodings are Reserved.

#### 5.5.1.4 SIZE

Bits[3:2] are reserved. Chipset drives them to 00. Peripherals running bus master cycles are also supposed to drive 00 for bits 3:2; however, Chipset ignores those bits. Bits[1:0] are encoded as listed in Table 5-36.

**Table 5-36.Transfer Size Bit Definition**

| Bits[1:0] | Size |
|---|---|
| 00 | 8-bit transfer (1 byte) |
| 01 | 16-bit transfer (2 bytes) |
| 10 | **Reserved**. Chipset does not drive this combination. If a peripheral running a bus master cycle drives this combination, Chipset may abort the transfer. |
| 11 | 32-bit transfer (4 bytes) |



#### 5.5.1.5 SYNC

Valid values for the SYNC field are shown in Table 5-37.

**Table 5-37.SYNC Bit Definition**

| Bits[3:0][1,2] | Indication |
|---|---|
| 0000 | **Ready:** SYNC achieved with no error. For DMA transfers, this also indicates DMA request deassertion and no more transfers desired for that channel. |
| 0101 | **Short Wait:** Part indicating wait-states. For bus master cycles, Chipset does not use this encoding. Instead, Chipset uses the Long Wait encoding (see next encoding below). |
| 0110 | **Long Wait:** Part indicating wait-states, and many wait-states will be added. This encoding driven by Chipset for bus master cycles, rather than the Short Wait (0101). |
| 1001 | **Ready More (Used only by peripheral for DMA cycle):** SYNC achieved with no error and more DMA transfers desired to continue after this transfer. This value is valid only on DMA transfers and is not allowed for any other type of cycle. |
| 1010 | **Error:** Sync achieved with error. This is generally used to replace the SERR# or IOCHK# signal on the PCI/ISA bus. It indicates that the data is to be transferred, but there is a serious error in this transfer. For DMA transfers, this not only indicates an error, but also indicates DMA request deassertion and no more transfers desired for that channel. |

**NOTES:**
1. All other combinations are Reserved.
2. If the LPC controller receives any SYNC returned from the device other than short (0101), long wait (0110), or ready (0000) when running a FWH cycle, indeterminate results may occur. A FWH device is not allowed to assert an Error SYNC.

#### 5.5.1.6 SYNC Time-Out

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

#### 5.5.1.7 SYNC Error Indication

Chipset responds as defined in section 4.2.1.10 of the *Low Pin Count Interface Specification,* Revision 1.1.

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

#### 5.5.1.8 LFRAME# Usage

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

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

- Chipset starts a Memory, I/O, or DMA cycle, but no device drives a valid SYNC after four consecutive clocks.
- Chipset starts a Memory, I/O, or DMA cycle, and the peripheral drives an invalid SYNC pattern.
- A peripheral drives an invalid address when performing bus master cycles.
- A peripheral drives an invalid value.

#### 5.5.1.9 I/O Cycles

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

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

#### 5.5.1.10 Bus Master Cycles

Chipset supports Bus Master cycles and requests (using LDRQ#) as defined in the *Low Pin Count Interface Specification,* Revision 1.1. Chipset has two LDRQ# inputs, and thus supports two separate bus master devices. It uses the associated START fields for Bus Master 0 (0010b) or Bus Master 1 (0011b).

***Note:*** Chipset does not support LPC Bus Masters performing I/O cycles. LPC Bus Masters should only perform memory read or memory write cycles.

#### 5.5.1.11 LPC Power Management

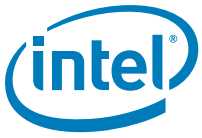
**CLKRUN# Protocol (Netbook)**

The CLKRUN# protocol is same as in the *PCI Local Bus Specification*. Stopping the PCI clock stops the LPC clock.

**LPCPD# Protocol**

Same timings as for SUS_STAT#. Upon driving SUS_STAT# low, LPC peripherals drive LDRQ# low or tri-state it. Chipset shuts off the LDRQ# input buffers. After driving SUS_STAT# active, Chipset drives LFRAME# low, and tri-states (or drive low) LAD[3:0].

***Note:*** The *Low Pin Count Interface Specification,* Revision 1.1 defines the LPCPD# protocol where there is at least 30 µs from LPCPD# assertion to LRST# assertion. This specification explicitly states that this protocol only applies to entry/exit of low power states which does not include asynchronous reset events. Chipset asserts both SUS_STAT# (connects to LPCPD#) and PLTRST# (connects to LRST#) at the same time



when the core logic is reset (via CF9h, PWROK, or SYS_RESET#, etc.). This is not inconsistent with the LPC LPCPD# protocol.

#### 5.5.1.12 Configuration and Chipset Implications

**LPC Interface Decoders**

To allow the I/O cycles and memory mapped cycles to go to the LPC interface, Chipset includes several decoders. During configuration, Chipset must be programmed with the same decode ranges as the peripheral. The decoders are programmed via the Device 31:Function 0 configuration space.

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

**Bus Master Device Mapping and START Fields**

Bus Masters must have a unique START field. In the case of Chipset that supports two LPC bus masters, it drives 0010 for the START field for grants to bus master #0 (requested via LDRQ0#) and 0011 for grants to bus master #1 (requested via LDRQ1#.). Thus, no registers are needed to configure the START fields for a particular bus master.

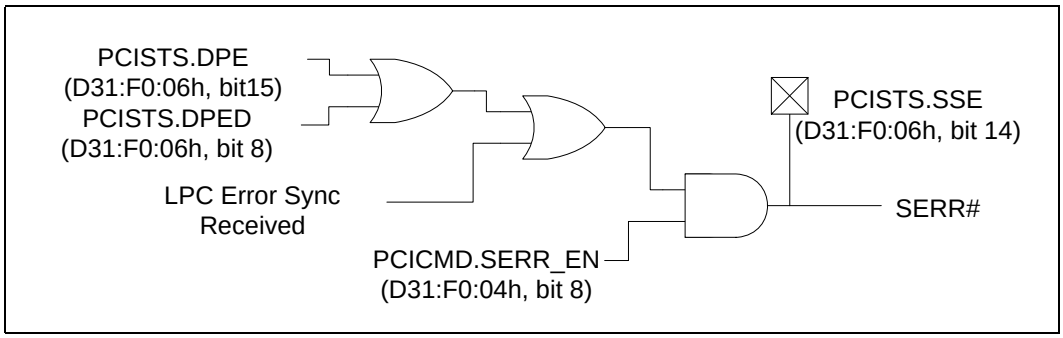
### 5.5.2 SERR# Generation

Several internal and external sources of the LPC Bridge can cause SERR#, as described below.

The first class of errors is parity errors related to the backbone. The LPC Bridge captures generic data parity errors (errors it finds on the backbone) as well as errors returned on the backbone cycles where the bridge was the master and parity error response is enabled. If either of these two conditions is met, and with SERR# enable (PCICMD.SERR_EN) set, SERR# will be captured.

Additionally, if the LPC Bridge receives an error SYNC on LPC bus, an SERR# will also be generated.

**Figure 5-9. LPC Bridge SERR# Generation**

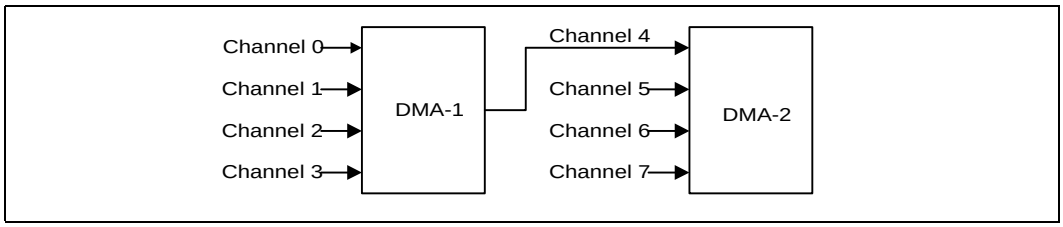



## 5.6 DMA Operation (D31:F0)

Chipset supports LPC DMA using chipset's DMA controller. The DMA controller has registers that are fixed in the lower 64 KB of I/O space. The DMA controller is configured using registers in the PCI configuration space. These registers allow configuration of the channels for use by LPC DMA.

The DMA circuitry incorporates the functionality of two 82C37 DMA controllers with seven independently programmable channels (Figure 5-10). DMA controller 1 (DMA-1) corresponds to DMA channels 0–3 and DMA controller 2 (DMA-2) corresponds to channels 5–7. DMA channel 4 is used to cascade the two controllers and defaults to cascade mode in the DMA Channel Mode (DCM) Register. Channel 4 is not available for any other purpose. In addition to accepting requests from DMA slaves, the DMA controller also responds to requests that software initiates. Software may initiate a DMA service request by setting any bit in the DMA Channel Request Register to a 1.

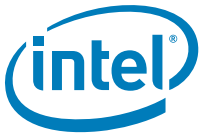
**Figure 5-10. Chipset DMA Controller**




Each DMA channel is hardwired to the compatible settings for DMA device size: channels [3:0] are hardwired to 8-bit, count-by-bytes transfers, and channels [7:5] are hardwired to 16-bit, count-by-words (address shifted) transfers.

Chipset provides 24-bit addressing in compliance with the ISA-Compatible specification. Each channel includes a 16-bit ISA-Compatible Current Register which holds the 16 least-significant bits of the 24-bit address, an ISA-Compatible Page Register which contains the eight next most significant bits of address.

The DMA controller also features refresh address generation, and autoinitialization following a DMA termination.



### 5.6.1 Channel Priority

For priority resolution, the DMA consists of two logical channel groups: channels 0–3 and channels 4–7. Each group may be in either fixed or rotate mode, as determined by the DMA Command Register.

DMA I/O slaves normally assert their DREQ line to arbitrate for DMA service. However, a software request for DMA service can be presented through each channel's DMA Request Register. A software request is subject to the same prioritization as any hardware request. See the detailed register description for Request Register programming information in Section 13.2.

#### 5.6.1.1 Fixed Priority

The initial fixed priority structure is as follows:

| High priority | Low priority |
|---|---|
| 0, 1, 2, 3 | 5, 6, 7 |

The fixed priority ordering is 0, 1, 2, 3, 5, 6, and 7. In this scheme, channel 0 has the highest priority, and channel 7 has the lowest priority. Channels [3:0] of DMA-1 assume the priority position of channel 4 in DMA-2, thus taking priority over channels 5, 6, and 7.

#### 5.6.1.2 Rotating Priority

Rotation allows for “fairness” in priority resolution. The priority chain rotates so that the last channel serviced is assigned the lowest priority in the channel group (0–3, 5–7).

Channels 0–3 rotate as a group of 4. They are placed between channel 5 and channel 7 in the priority list.

Channel 5–7 rotate as part of a group of 4. That is, channels (5–7) form the first three positions in the rotation, while channel group (0–3) comprises the fourth position in the arbitration.

### 5.6.2 Address Compatibility Mode

When the DMA is operating, the addresses do not increment or decrement through the High and Low Page Registers. Therefore, if a 24-bit address is 01FFFFh and increments, the next address is 010000h, not 020000h. Similarly, if a 24-bit address is 020000h and decrements, the next address is 02FFFFh, not 01FFFFh. However, when the DMA is operating in 16-bit mode, the addresses still do not increment or decrement through the High and Low Page Registers but the page boundary is now 128 K. Therefore, if a 24-bit address is 01FFFEh and increments, the next address is 000000h, not 0100000h. Similarly, if a 24-bit address is 020000h and decrements, the next address is 03FFFEh, not 02FFFEh. This is compatible with the 82C37 and Page Register implementation used in the PC-AT. This mode is set after CPURST is valid.

## 5.6.3 Summary of DMA Transfer Sizes

Table 5-38 lists each of the DMA device transfer sizes. The column labeled “Current Byte/Word Count Register” indicates that the register contents represents either the number of bytes to transfer or the number of 16-bit words to transfer. The column labeled “Current Address Increment/Decrement” indicates the number added to or taken from the Current Address register after each DMA transfer cycle. The DMA Channel Mode Register determines if the Current Address Register will be incremented or decremented.

### 5.6.3.1 Address Shifting When Programmed for 16-Bit I/O Count by Words

**Table 5-38. DMA Transfer Size**

| DMA Device Date Size And Word Count | Current Byte/Word Count Register | Current Address Increment/ Decrement |
|---|---|---|
| 8-Bit I/O, Count By Bytes | Bytes | 1 |
| 16-Bit I/O, Count By Words (Address Shifted) | Words | 1 |

Chipset maintains compatibility with the implementation of the DMA in the PC AT that used the 82C37. The DMA shifts the addresses for transfers to/from a 16-bit device count-by-words.

***Note:*** The least significant bit of the Low Page Register is dropped in 16-bit shifted mode. When programming the Current Address Register (when the DMA channel is in this mode), the Current Address must be programmed to an even address with the address value shifted right by one bit.

The address shifting is shown in Table 5-39.

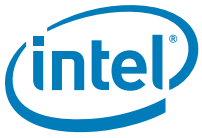
**Table 5-39. Address Shifting in 16-Bit I/O DMA Transfers**

| Output Address | 8-Bit I/O Programmed Address (Ch 0–3) | 16-Bit I/O Programmed Address (Ch 5–7) (Shifted) |
|---|---|---|
| A0<br>A[16:1]<br>A[23:17] | A0<br>A[16:1]<br>A[23:17] | 0<br>A[15:0]<br>A[23:17] |

**NOTE:** The least significant bit of the Page Register is dropped in 16-bit shifted mode.

## 5.6.4 Autoinitialize

By programming a bit in the DMA Channel Mode Register, a channel may be set up as an autoinitialize channel. When a channel undergoes autoinitialization, the original values of the Current Page, Current Address and Current Byte/Word Count Registers are automatically restored from the Base Page, Address, and Byte/Word Count Registers of that channel following TC. The Base Registers are loaded simultaneously with the Current Registers by the microprocessor when the DMA channel is



programmed and remain unchanged throughout the DMA service. The mask bit is not set when the channel is in autoinitialize. Following autoinitialize, the channel is ready to perform another DMA service, without processor intervention, as soon as a valid DREQ is detected.

### 5.6.5 Software Commands

There are three additional special software commands that the DMA controller can execute. The three software commands are:

- Clear Byte Pointer Flip-Flop
- Master Clear
- Clear Mask Register

They do not depend on any specific bit pattern on the data bus.

## 5.7 LPC DMA

DMA on LPC is handled through the use of the LDRQ# lines from peripherals and special encodings on LAD[3:0] from the host. Single, Demand, Verify, and Increment modes are supported on the LPC interface. Channels 0–3 are 8 bit channels. Channels 5–7 are 16-bit channels. Channel 4 is reserved as a generic bus master request.

### 5.7.1 Asserting DMA Requests

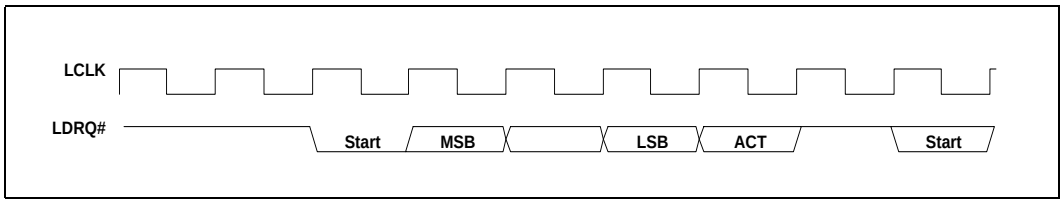
Peripherals that need DMA service encode their requested channel number on the LDRQ# signal. To simplify the protocol, each peripheral on the LPC I/F has its own dedicated LDRQ# signal (they may not be shared between two separate peripherals). Chipset has two LDRQ# inputs, allowing at least two devices to support DMA or bus mastering.

LDRQ# is synchronous with LCLK (PCI clock). As shown in Figure 5-11, the peripheral uses the following serial encoding sequence:

- Peripheral starts the sequence by asserting LDRQ# low (start bit). LDRQ# is high during idle conditions.
- The next three bits contain the encoded DMA channel number (MSB first).
- The next bit (ACT) indicates whether the request for the indicated DMA channel is active or inactive. The ACT bit is 1 (high) to indicate if it is active and 0 (low) if it is inactive. The case where ACT is low is rare, and is only used to indicate that a previous request for that channel is being abandoned.
- After the active/inactive indication, the LDRQ# signal must go high for at least 1 clock. After that one clock, LDRQ# signal can be brought low to the next encoding sequence.

If another DMA channel also needs to request a transfer, another sequence can be sent on LDRQ#. For example, if an encoded request is sent for channel 2, and then channel 3 needs a transfer before the cycle for channel 2 is run on the interface, the peripheral can send the encoded request for channel 3. This allows multiple DMA agents behind an I/O device to request use of the LPC interface, and the I/O device does not need to self-arbitrate before sending the message.

**Figure 5-11. DMA Request Assertion through LDRQ#**




## 5.7.2 Abandoning DMA Requests

DMA Requests can be deasserted in two fashions: on error conditions by sending an LDRQ# message with the 'ACT' bit set to 0, or normally through a SYNC field during the DMA transfer. This section describes boundary conditions where the DMA request needs to be removed prior to a data transfer.

There may be some special cases where the peripheral desires to abandon a DMA transfer. The most likely case of this occurring is due to a floppy disk controller which has overrun or underrun its FIFO, or software stopping a device prematurely.

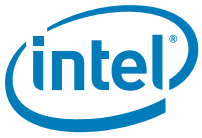
In these cases, the peripheral wishes to stop further DMA activity. It may do so by sending an LDRQ# message with the ACT bit as 0. However, since the DMA request was seen by Chipset, there is no assurance that the cycle has not been granted and will shortly run on LPC. Therefore, peripherals must take into account that a DMA cycle may still occur. The peripheral can choose not to respond to this cycle, in which case the host will abort it, or it can choose to complete the cycle normally with any random data.

This method of DMA deassertion should be prevented whenever possible, to limit boundary conditions both on Chipset and the peripheral.

## 5.7.3 General Flow of DMA Transfers

Arbitration for DMA channels is performed through the 8237 within the host. Once the host has won arbitration on behalf of a DMA channel assigned to LPC, it asserts LFRAME# on the LPC I/F and begins the DMA transfer. The general flow for a basic DMA transfer is as follows:

1. Chipset starts transfer by asserting 0000b on LAD[3:0] with LFRAME# asserted.
2. Chipset asserts 'cycle type' of DMA, direction based on DMA transfer direction.
3. Chipset asserts channel number and, if applicable, terminal count.
4. Chipset indicates the size of the transfer: 8 or 16 bits.
5. If a DMA read -



- — Chipset drives the first 8 bits of data and turns the bus around.
- — The peripheral acknowledges the data with a valid SYNC.
- — If a 16-bit transfer, the process is repeated for the next 8 bits.

6. If a DMA write -
   - — Chipset turns the bus around and waits for data.
   - — The peripheral indicates data ready through SYNC and transfers the first byte.
   - — If a 16-bit transfer, the peripheral indicates data ready and transfers the next byte.
7. The peripheral turns around the bus.

### 5.7.4 Terminal Count

Terminal count is communicated through LAD[3] on the same clock that DMA channel is communicated on LAD[2:0]. This field is the CHANNEL field. Terminal count indicates the last byte of transfer, based upon the size of the transfer.

For example, on an 8-bit transfer size (SIZE field is 00b), if the TC bit is set, then this is the last byte. On a 16-bit transfer (SIZE field is 01b), if the TC bit is set, then the second byte is the last byte. The peripheral, therefore, must internalize the TC bit when the CHANNEL field is communicated, and only signal TC when the last byte of that transfer size has been transferred.

### 5.7.5 Verify Mode

Verify mode is supported on the LPC interface. A verify transfer to the peripheral is similar to a DMA write, where the peripheral is transferring data to main memory. The indication from the host is the same as a DMA write, so the peripheral will be driving data onto the LPC interface. However, the host will not transfer this data into main memory.

### 5.7.6 DMA Request Deassertion

An end of transfer is communicated to Chipset through a special SYNC field transmitted by the peripheral. An LPC device must not attempt to signal the end of a transfer by deasserting LDREQ#. If a DMA transfer is several bytes (e.g., a transfer from a demand mode device) Chipset needs to know when to deassert the DMA request based on the data currently being transferred.

The DMA agent uses a SYNC encoding on each byte of data being transferred, which indicates to Chipset whether this is the last byte of transfer or if more bytes are requested. To indicate the last byte of transfer, the peripheral uses a SYNC value of 0000b (ready with no error), or 1010b
(ready with error). These encodings tell Chipset that this is the last piece of data transferred on a DMA read (Chipset to peripheral), or the byte that follows is the last piece of data transferred on a DMA write (peripheral to Chipset).

When Chipset sees one of these two encodings, it ends the DMA transfer after this byte and deasserts the DMA request to the 8237. Therefore, if Chipset indicated a 16-bit transfer, the peripheral can end the transfer after one byte by indicating a SYNC value of 0000b or 1010b. Chipset does not attempt to transfer the second byte, and deasserts the DMA request internally.

If the peripheral indicates a 0000b or 1010b SYNC pattern on the last byte of the indicated size, then Chipset only deasserts the DMA request to the 8237 since it does not need to end the transfer.

If the peripheral wishes to keep the DMA request active, then it uses a SYNC value of 1001b (ready plus more data). This tells the 8237 that more data bytes are requested after the current byte has been transferred, so Chipset keeps the DMA request active to the 8237. Therefore, on an 8-bit transfer size, if the peripheral indicates a SYNC value of 1001b to Chipset, the data will be transferred and the DMA request will remain active to the 8237. At a later time, Chipset will then come back with another START–CYCTYPE–CHANNEL–SIZE etc. combination to initiate another transfer to the peripheral.

The peripheral must not assume that the next START indication from Chipset is another grant to the peripheral if it had indicated a SYNC value of 1001b. On a single mode DMA device, the 8237 will re-arbitrate after every transfer. Only demand mode DMA devices can be ensured that they will receive the next START indication from Chipset.

***Note:*** Indicating a 0000b or 1010b encoding on the SYNC field of an odd byte of a 16-bit channel (first byte of a 16-bit transfer) is an error condition.

***Note:*** The host stops the transfer on the LPC bus as indicated, fills the upper byte with random data on DMA writes (peripheral to memory), and indicates to the 8237 that the DMA transfer occurred, incrementing the 8237's address and decrementing its byte count.

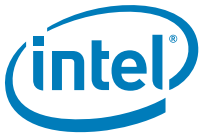
### 5.7.7 SYNC Field / LDRQ# Rules

Since DMA transfers on LPC are requested through an LDRQ# assertion message, and are ended through a SYNC field during the DMA transfer, the peripheral must obey the following rule when initiating back-to-back transfers from a DMA channel.

The peripheral must not assert another message for eight LCLKs after a deassertion is indicated through the SYNC field. This is needed to allow the 8237, that typically runs off a much slower internal clock, to see a message deasserted before it is re-asserted so that it can arbitrate to the next agent.

Under default operation, the host only performs 8-bit transfers on 8-bit channels and 16-bit transfers on 16-bit channels.

The method by which this communication between host and peripheral through system BIOS is performed is beyond the scope of this specification. Since the LPC host and LPC peripheral are motherboard devices, no "plug-n-play" registry is required.



The peripheral must not assume that the host is able to perform transfer sizes that are larger than the size allowed for the DMA channel, and be willing to accept a SIZE field that is smaller than what it may currently have buffered.

To that end, it is recommended that future devices that may appear on the LPC bus, that require higher bandwidth than 8-bit or 16-bit DMA allow, do so with a bus mastering interface and not rely on the 8237.

## 5.8 8254 Timers (D31:F0)

Chipset contains three counters that have fixed uses. All registers and functions associated with the 8254 timers are in the core well. The 8254 unit is clocked by a 14.31818 MHz 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 1, Refresh Request Signal

This counter provides the refresh request signal and is typically programmed for Mode 2 operation and only impacts the period of the REF_TOGGLE bit in Port 61. The initial count value is loaded one counter period after being written to the counter I/O address. The REF_TOGGLE bit will have a square wave behavior (alternate between 0 and 1) and will toggle at a rate based on the value in the counter. Programming the counter to anything other than Mode 2 will result in undefined behavior for the REF_TOGGLE bit.

### Counter 2, Speaker Tone

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

### 5.8.1 Timer Programming

The counter/timers are programmed in the following fashion:

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

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

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

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

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

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

Table 5-40 lists the six operating modes for the interval counters.

**Table 5-40.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. |

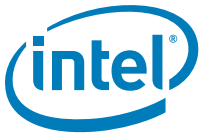


**Table 5-40.Counter Operating Modes**

| | | |
|---|---|---|
| 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, etc. |
| 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. |

## 5.8.2 Reading from the Interval Timer

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

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

### 5.8.2.1 Simple Read

The first method is to perform a simple read operation. The counter is selected through port 40h (counter 0), 41h (counter 1), 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.

### 5.8.2.2 Counter Latch Command

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

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

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

#### 5.8.2.3 Read Back Command

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

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

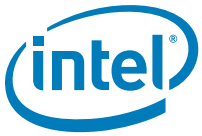
The Read Back command may additionally be used to latch status information of selected counters. The status of a counter is accessed by a read from that counter's I/O port address. If multiple counter status latch operations are performed without reading the status, all but the first are ignored.

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

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

## 5.9 8259 Interrupt Controllers (PIC) (D31:F0)

Chipset incorporates the functionality of two 8259 interrupt controllers that provide system interrupts for the ISA compatible interrupts. These interrupts are: system timer, keyboard controller, serial ports, parallel ports, floppy disk, IDE, mouse, and DMA channels. In addition, this interrupt controller can support the PCI based interrupts, by mapping the PCI interrupt onto the compatible ISA interrupt line. Each 8259 core supports eight interrupts, numbered 0–7. Table 5-41 shows how the cores are connected.



### Table 5-41. Interrupt Controller Core Connections

| 8259 | 8259 Input | Typical Interrupt Source | Connected Pin / Function |
|---|---|---|---|
| Master | 0 | Internal | Internal Timer / Counter 0 output / HPET #0 |
| | 1 | Keyboard | IRQ1 via SERIRQ |
| | 2 | Internal | Slave controller INTR output |
| | 3 | Serial Port A | IRQ3 via SERIRQ, PIRQ# |
| | 4 | Serial Port B | IRQ4 via SERIRQ, PIRQ# |
| | 5 | Parallel Port / Generic | IRQ5 via SERIRQ, PIRQ# |
| | 6 | Floppy Disk | IRQ6 via SERIRQ, PIRQ# |
| | 7 | Parallel Port / Generic | IRQ7 via SERIRQ, PIRQ# |
| Slave | 0 | Internal Real Time Clock | Internal RTC / HPET #1 |
| | 1 | Generic | IRQ9 via SERIRQ, SCI, TCO, or PIRQ# |
| | 2 | Generic | IRQ10 via SERIRQ, SCI, TCO, or PIRQ# |
| | 3 | Generic | IRQ11 via SERIRQ, SCI, TCO, or PIRQ# |
| | 4 | PS/2 Mouse | IRQ12 via SERIRQ, SCI, TCO, or PIRQ# |
| | 5 | Internal | State Machine output based on processor FERR# assertion. May optionally be used for SCI or TCO interrupt if FERR# not needed. |
| | 6 | SATA | SATA Primary (legacy mode), or via SERIRQ or PIRQ# |
| | 7 | SATA | SATA Secondary (legacy mode) or via SERIRQ or PIRQ# |

Chipset cascades the slave controller onto the master controller through master controller interrupt input 2. This means there are only 15 possible interrupts for Chipset PIC.

Interrupts can individually be programmed to be edge or level, except for IRQ0, IRQ2, IRQ8#, and IRQ13.

***Note:*** Active-low interrupt sources (e.g., the PIRQ#s) are inverted inside Chipset. In the following descriptions of the 8259s, the interrupt levels are in reference to the signals at the internal interface of the 8259s, after the required inversions have occurred. Therefore, the term “high” indicates “active,” which means “low” on an originating PIRQ#.

## 5.9.1 Interrupt Handling

### 5.9.1.1 Generating Interrupts

The PIC interrupt sequence involves three bits, from the IRR, ISR, and IMR, for each interrupt level. These bits are used to determine the interrupt vector returned, and status of any other pending interrupts. Table 5-42 defines the IRR, ISR, and IMR.

**Table 5-42. Interrupt Status Registers**

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

### 5.9.1.2 Acknowledging Interrupts

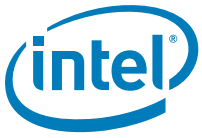
The processor generates an interrupt acknowledge cycle that is translated by the host bridge into a PCI Interrupt Acknowledge Cycle to Chipset. The PIC translates this command into two internal INTA# pulses expected by the 8259 cores. The PIC uses the first internal INTA# pulse to freeze the state of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

**Table 5-43. Content of Interrupt Vector Byte**

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

### 5.9.1.3 Hardware/Software Interrupt Sequence

1. One or more of the Interrupt Request lines (IRQ) are raised high in edge mode, or seen high in level mode, setting the corresponding IRR bit.
2. The PIC sends INTR active to the processor if an asserted interrupt is not masked.



3. The processor acknowledges the INTR and responds with an interrupt acknowledge cycle. The cycle is translated into a PCI interrupt acknowledge cycle by the host bridge. This command is broadcast over PCI by Chipset.
4. Upon observing its own interrupt acknowledge cycle on PCI, Chipset converts it into the two cycles that the internal 8259 pair can respond to. Each cycle appears as an interrupt acknowledge pulse on the internal INTA# pin of the cascaded interrupt controllers.
5. Upon receiving the first internally generated INTA# pulse, the highest priority ISR bit is set and the corresponding IRR bit is reset. On the trailing edge of the first pulse, a slave identification code is broadcast by the master to the slave on a private, internal three bit wide bus. The slave controller uses these bits to determine if it must respond with an interrupt vector during the second INTA# pulse.
6. Upon receiving the second internally generated INTA# pulse, the PIC returns the interrupt vector. If no interrupt request is present because the request was too short in duration, the PIC returns vector 7 from the master controller.
7. This completes the interrupt cycle. In AEOI mode the ISR bit is reset at the end of the second INTA# pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

## 5.9.2 Initialization Command Words (ICWx)

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

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

### 5.9.2.1 ICW1

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

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

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

#### 5.9.2.2 ICW2

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

#### 5.9.2.3 ICW3

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

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

#### 5.9.2.4 ICW4

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

### 5.9.3 Operation Command Words (OCW)

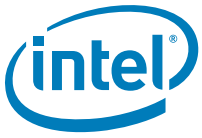
These command words reprogram the Interrupt controller to operate in various interrupt modes.

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

### 5.9.4 Modes of Operation

#### 5.9.4.1 Fully Nested Mode

In this mode, interrupt requests are ordered in priority from 0 through 7, with 0 being the highest. When an interrupt is acknowledged, the highest priority request is determined and its vector placed on the bus. Additionally, the ISR for the interrupt is set. This ISR bit remains set until: the processor issues an EOI command immediately before returning from the service routine; or if in AEOI mode, on the trailing edge of



the second INTA#. While the ISR bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels generate another interrupt. Interrupt priorities can be changed in the rotating priority mode.

#### 5.9.4.2 Special Fully-Nested Mode

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

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

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

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

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

#### 5.9.4.4 Specific Rotation Mode (Specific Priority)

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

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

#### 5.9.4.5 Poll Mode

Poll mode can be used to conserve space in the interrupt vector table. Multiple interrupts that can be serviced by one interrupt service routine do not need separate vectors if the service routine uses the poll command. Poll mode can also be used to

expand the number of interrupts. The polling interrupt service routine can call the appropriate service routine, instead of providing the interrupt vectors in the vector table. In this mode, the INTR output is not used and the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by software using a Poll command.

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

#### 5.9.4.6 Cascade Mode

The PIC in Chipset has one master 8259 and one slave 8259 cascaded onto the master through IRQ2. This configuration can handle up to 15 separate priority levels. The master controls the slaves through a three bit internal bus. In Chipset, when the master drives 010b on this bus, the slave controller takes responsibility for returning the interrupt vector. An EOI command must be issued twice: once for the master and once for the slave.

#### 5.9.4.7 Edge and Level Triggered Mode

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

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

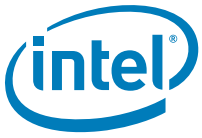
In both the edge and level triggered modes, the IRQ inputs must remain active until after the falling edge of the first internal INTA#. If the IRQ input goes inactive before this time, a default IRQ7 vector is returned.

#### 5.9.4.8 End of Interrupt (EOI) Operations

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

#### 5.9.4.9 Normal End of Interrupt

In normal EOI, software writes an EOI command before leaving the interrupt service routine to mark the interrupt as completed. There are two forms of EOI commands: Specific and



Non-Specific. When a Non-Specific EOI command is issued, the PIC clears the highest ISR bit of those that are set to 1. Non-Specific EOI is the normal mode of operation of the PIC within Chipset, as the interrupt being serviced currently is the interrupt entered with the interrupt acknowledge. When the PIC is operated in modes that preserve the fully nested structure, software can determine which ISR bit to clear by issuing a Specific EOI. An ISR bit that is masked is not cleared by a Non-Specific EOI if the PIC is in the special mask mode. An EOI command must be issued for both the master and slave controller.

#### 5.9.4.10 Automatic End of Interrupt Mode

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

### 5.9.5 Masking Interrupts

#### 5.9.5.1 Masking on an Individual Interrupt Request

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

#### 5.9.5.2 Special Mask Mode

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

The special mask mode enables all interrupts not masked by a bit set in the Mask register. Normally, when an interrupt service routine acknowledges an interrupt without issuing an EOI to clear the ISR bit, the interrupt controller inhibits all lower priority requests. In the special mask mode, any interrupts may be selectively enabled by loading the Mask Register with the appropriate pattern. The special mask mode is set by OCW3 where: SSMM=1, SMM=1, and cleared where SSMM=1, SMM=0.

### 5.9.6 Steering PCI Interrupts

Chipset can be programmed to allow PIRQA#-PIRQH# to be internally routed to interrupts 3–7, 9–12, 14 or 15. The assignment is programmable through the through the PIRQx Route Control registers, located at 60–63h and 68–6Bh in Device 31:Function 0. One or more PIRQx# lines can be routed to the same IRQx input. If interrupt steering is not required, the Route registers can be programmed to disable steering.

The PIRQx# lines are defined as active low, level sensitive to allow multiple interrupts on a PCI board to share a single line across the connector. When a PIRQx# is routed to specified IRQ line, software must change the IRQ's corresponding ELCR bit to level sensitive mode. Chipset internally inverts the PIRQx# line to send an active high level to the PIC. When a PCI interrupt is routed onto the PIC, the selected IRQ can no longer be used by an active high device (through SERIRQ). However, active low interrupts can share their interrupt with PCI interrupts.

Internal sources of the PIRQs, including SCI and TCO interrupts, cause the external PIRQ to be asserted. Chipset receives the PIRQ input, like all of the other external sources, and routes it accordingly.

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

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

### 5.10.1 Interrupt Handling

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

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

### 5.10.2 Interrupt Mapping

The I/O APIC within Chipset supports 24 APIC interrupts. Each interrupt has its own unique vector assigned by software. The interrupt vectors are mapped as follows, and match “Config 6” of the *Multi-Processor Specification*.

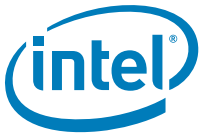


**Table 5-44.APIC Interrupt Mapping**

| IRQ #[1] | Via SERIRQ | Direct from Pin | Via PCI Message | Internal Modules |
|---|---|---|---|---|
| 0 | No | No | No | Cascade from 8259 #1 |
| 1 | Yes | No | Yes | |
| 2 | No | No | No | 8254 Counter 0, HPET #0 (legacy mode) |
| 3 | Yes | No | Yes | |
| 4 | Yes | No | Yes | |
| 5 | Yes | No | Yes | |
| 6 | Yes | No | Yes | |
| 7 | Yes | No | Yes | |
| 8 | No | No | No | RTC, HPET #1 (legacy mode) |
| 9 | Yes | No | Yes | Option for SCI, TCO |
| 10 | Yes | No | Yes | Option for SCI, TCO |
| 11 | Yes | No | Yes | HPET #2, Option for SCI, TCO[2] |
| 12 | Yes | No | Yes | |
| 13 | No | No | No | FERR# logic |
| 14 | Yes | Yes[3] | Yes | SATA Primary (legacy mode) |
| 15 | Yes | Yes | Yes | SATA Secondary (legacy mode) |
| 16 | PIRQA# | PIRQA# | Yes | Internal devices are routable; see Section 10.1.37 though Section 10.1.46. |
| 17 | PIRQB# | PIRQB# | | |
| 18 | PIRQC# | PIRQC# | | |
| 19 | PIRQD# | PIRQD# | | |
| 20 | N/A | PIRQE# | Yes | Option for SCI, TCO, HPET #0,1,2. Other internal devices are routable; see Section 10.1.37 though Section 10.1.46. |
| 21 | N/A | PIRQF# | | |
| 22 | N/A | PIRQG# | | |
| 23 | N/A | PIRQH# | | |

**NOTES:**
1. When programming the polarity of internal interrupt sources on the APIC, interrupts 0 through 15 receive active-high internal interrupt sources, while interrupts 16 through 23 receive active-low internal interrupt sources.
2. If IRQ 11 is used for HPET #2, software should ensure IRQ 11 is not shared with any other devices to ensure the proper operation of HPET #2. Chipset hardware does not prevent sharing of IRQ 11.

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

When external devices through PCI / PCI Express wish to generate an interrupt, they will send the message defined in the *PCI Express* Base Specification,* Revision 1.0a for generating INTA# – INTD#. These will be translated internal assertions/de-assertions of INTA# – INTD#.

### 5.10.4 Front Side Bus Interrupt Delivery

For processors that support Front Side Bus (FSB) interrupt delivery, Chipset requires that the I/O APIC deliver interrupt messages to the processor in a parallel manner, rather than using the I/O APIC serial scheme.

This is done by Chipset writing (via DMI) to a memory location that is snooped by the processor(s). The processor(s) snoop the cycle to know which interrupt goes active.

The following sequence is used:

1. When Chipset detects an interrupt event (active edge for edge-triggered mode or a change for level-triggered mode), it sets or resets the internal IRR bit associated with that interrupt.
2. Internally, Chipset requests to use the bus in a way that automatically flushes upstream buffers. This can be internally implemented similar to a DMA device request.
3. Chipset then delivers the message by performing a write cycle to the appropriate address with the appropriate data. The address and data formats are described below in Section 5.10.4.4.

***Note:*** FSB Interrupt Delivery compatibility with processor clock control depends on the processor, not Chipset.

#### 5.10.4.1 Edge-Triggered Operation

In this case, the “Assert Message” is sent when there is an inactive-to-active edge on the interrupt.

#### 5.10.4.2 Level-Triggered Operation

In this case, the “Assert Message” is sent when there is an inactive-to-active edge on the interrupt. If after the EOI the interrupt is still active, then another “Assert Message” is sent to indicate that the interrupt is still active.

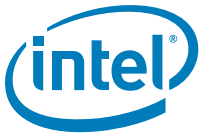
#### 5.10.4.3 Registers Associated with Front Side Bus Interrupt Delivery

**Capabilities Indication:** The capability to support Front Side Bus interrupt delivery is indicated via ACPI configuration techniques. This involves the BIOS creating a data structure that gets reported to the ACPI configuration software.

#### 5.10.4.4 Interrupt Message Format

Chipset writes the message to PCI (and to the Host controller) as a 32-bit memory write cycle. It uses the formats shown in Table 5-45 and Table 5-46 for the address and data.

The local APIC (in the processor) has a delivery mode option to interpret Front Side Bus messages as a SMI in which case the processor treats the incoming interrupt as a SMI instead of as an interrupt. This does not mean that Chipset has any way to have a SMI



source from Chipset power management logic cause the I/O APIC to send an SMI message (there is no way to do this). chipset's I/O APIC can only send interrupts due to interrupts which do not include SMI, NMI or INIT. This means that in IA32/IA64 based platforms, Front Side Bus interrupt message format delivery modes 010 (SMI/PMI), 100 (NMI), and 101 (INIT) as indicated in this section, must not be used and is not supported. Only the hardware pin connection is supported by Chipset.

### Table 5-45. Interrupt Message Address Format

| Bit | Description |
|---|---|
| 31:20 | Will always be FEEh |
| 19:12 | **Destination ID:** This is the same as bits 63:56 of the I/O Redirection Table entry for the interrupt associated with this message. |
| 11:4 | **Extended Destination ID**: This is the same as bits 55:48 of the I/O Redirection Table entry for the interrupt associated with this message. |
| 3 | **Redirection Hint:** This bit is used by the processor host bridge to allow the interrupt message to be redirected.<br>0 = The message will be delivered to the agent (processor) listed in bits 19:12.<br>1 = The message will be delivered to an agent with a lower interrupt priority This can be derived from bits 10:8 in the Data Field (see below).<br>The Redirection Hint bit will be a 1 if bits 10:8 in the delivery mode field associated with corresponding interrupt are encoded as 001 (Lowest Priority). Otherwise, the Redirection Hint bit will be 0 |
| 2 | **Destination Mode:** This bit is used only the Redirection Hint bit is set to 1. If the Redirection Hint bit and the Destination Mode bit are both set to 1, then the logical destination mode is used, and the redirection is limited only to those processors that are part of the logical group as based on the logical ID. |
| 1:0 | Will always be 00. |

### Table 5-46. Interrupt Message Data Format

| Bit | Description |
|---|---|
| 31:16 | Will always be 0000h. |
| 15 | **Trigger Mode:** 1 = Level, 0 = Edge. Same as the corresponding bit in the I/O Redirection Table for that interrupt. |
| 14 | **Delivery Status:** 1 = Assert, 0 = Deassert. Only Assert messages are sent. This bit is always 1. |
| 13:12 | Will always be 00 |
| 11 | **Destination Mode:** 1 = Logical. 0 = Physical. Same as the corresponding bit in the I/O Redirection Table for that interrupt. |
| 10:8 | **Delivery Mode:** This is the same as the corresponding bits in the I/O Redirection Table for that interrupt.<br>000 = Fixed 100 = NMI<br>001 = Lowest Priority 101 = INIT<br>010 = SMI/PMI 110 = **Reserved**<br>011 = **Reserved** 111 = ExtINT |
| 7:0 | **Vector:** This is the same as the corresponding bits in the I/O Redirection Table for that interrupt. |

## 5.11 Serial Interrupt (D31:F0)

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

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

Chipset supports a message for 21 serial interrupts. These represent the 15 ISA interrupts (IRQ0–1, 2–15), the four PCI interrupts, and the control signals SMI# and IOCHK#. The serial IRQ protocol does not support the additional APIC interrupts (20–23).

### 5.11.1 Start Frame

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

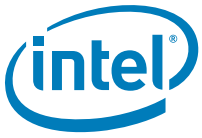
The mode that must first be entered when enabling the serial IRQ protocol is continuous mode. In this mode, Chipset asserts the start frame. This start frame is 4, 6, or 8 PCI clocks wide based upon the Serial IRQ Control Register, bits 1:0 at 64h in Device 31:Function 0 configuration space. This is a polling mode.

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

### 5.11.2 Data Frames

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

- **Sample Phase.** During this phase, the SERIRQ device drives SERIRQ low if the corresponding interrupt signal is low. If the corresponding interrupt is high, then the SERIRQ devices tri-state the SERIRQ signal. The SERIRQ line remains high due to pull-up resistors (there is no internal pull-up resistor on this signal, an external



pull-up resistor is required). A low level during the IRQ0–1 and IRQ2–15 frames indicates that an active-high ISA interrupt is not being requested, but a low level during the PCI INT[A:D], SMI#, and IOCHK# frame indicates that an active-low interrupt is being requested.

- **Recovery Phase.** During this phase, the device drives the SERIRQ line high if in the Sample Phase it was driven low. If it was not driven in the sample phase, it is tri-stated in this phase.
- **Turn-around Phase.** The device tri-states the SERIRQ line

### 5.11.3 Stop Frame

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

**Table 5-47.Stop Frame Explanation**

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

### 5.11.4 Specific Interrupts Not Supported via SERIRQ

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

- IRQ0. Heartbeat interrupt generated off of the internal 8254 counter 0.
- IRQ8#. RTC interrupt can only be generated internally.
- IRQ13. Floating point error interrupt generated off of the processor assertion of FERR#.

Chipset ignores the state of these interrupts in the serial stream, and does not adjust their level based on the level seen in the serial stream.

### 5.11.5 Data Frame Format

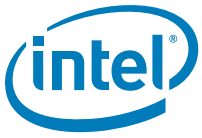
Table 5-48 shows the format of the data frames. For the PCI interrupts (A–D), the output from Chipset is ANDed with the PCI input signal. This way, the interrupt can be signaled via both the PCI interrupt input signal and via the SERIRQ signal (they are shared).

**Table 5-48.Data Frame Format**

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

## 5.12 Real Time Clock (D31:F0)

The Real Time Clock (RTC) module provides a battery backed-up date and time keeping device with two banks of static RAM with 128 bytes each, although the first bank has 114 bytes for general purpose usage. Three interrupt features are available: time of day alarm with once a second to once a month range, periodic rates of 122 µs to 500 ms, and end of update cycle notification. Seconds, minutes, hours, days, day of week, month, and year are counted. Daylight savings compensation is available. The hour is represented in twelve or twenty-four hour format, and data can be represented in BCD or binary format. The design is functionally compatible with the Motorola MS146818B. The time keeping comes from a 32.768 kHz oscillating source, which is divided to achieve an update every second. The lower 14 bytes on the lower RAM block has very specific functions. The first ten are for time and date information. The next four (0Ah to 0Dh) are registers, which configure and report RTC functions.



The time and calendar data should match the data mode (BCD or binary) and hour mode (12 or 24 hour) as selected in register B. It is up to the programmer to make sure that data stored in these locations is within the reasonable values ranges and represents a possible date and time. The exception to these ranges is to store a value of C0–FFh in the Alarm bytes to indicate a don't care situation. All Alarm conditions must match to trigger an Alarm Flag, which could trigger an Alarm Interrupt if enabled. The SET bit must be 1 while programming these locations to avoid clashes with an update cycle. Access to time and date information is done through the RAM locations. If a RAM read from the ten time and date bytes is attempted during an update cycle, the value read do not necessarily represent the true contents of those locations. Any RAM writes under the same conditions are ignored.

***Note:*** The leap year determination for adding a 29th day to February does not take into account the end-of-the-century exceptions. The logic simply assumes that all years divisible by 4 are leap years. According to the Royal Observatory Greenwich, years that are divisible by 100 are typically not leap years. In every fourth century (years divisible by 400, like 2000), the 100-year-exception is over-ridden and a leap-year occurs. Note that the year 2100 will be the first time in which the current RTC implementation would incorrectly calculate the leap-year.

Chipset does not implement month/year alarms.

### 5.12.1 Update Cycles

An update cycle occurs once a second, if the SET bit of register B is not asserted and the divide chain is properly configured. During this procedure, the stored time and date are incremented, overflow is checked, a matching alarm condition is checked, and the time and date are rewritten to the RAM locations. The update cycle will start at least 488 µs after the UIP bit of register A is asserted, and the entire cycle does not take more than 1984 µs to complete. The time and date RAM locations (0–9) are disconnected from the external bus during this time.

To avoid update and data corruption conditions, external RAM access to these locations can safely occur at two times. When a updated-ended interrupt is detected, almost 999 ms is available to read and write the valid time and date data. If the UIP bit of Register A is detected to be low, there is at least 488 µs before the update cycle begins.

***Warning:*** The overflow conditions for leap years and daylight savings adjustments are based on more than one date or time item. To ensure proper operation when adjusting the time, the new time and data values should be set at least two seconds before one of these conditions (leap year, daylight savings time adjustments) occurs.

### 5.12.2 Interrupts

The real-time clock interrupt is internally routed within Chipset both to the I/O APIC and the 8259. It is mapped to interrupt vector 8. This interrupt does not leave Chipset, nor is it shared with any other interrupt. IRQ8# from the SERIRQ stream is ignored. However, the High Performance Event Timers can also be mapped to IRQ8#; in this case, the RTC interrupt is blocked.

### 5.12.3 Lockable RAM Ranges

The RTC's battery-backed RAM supports two 8-byte ranges that can be locked via the configuration space. If the locking bits are set, the corresponding range in the RAM will not be readable or writable. A write cycle to those locations will have no effect. A read cycle to those locations will not return the location's actual value (resultant value is undefined).

Once a range is locked, the range can be unlocked only by a hard reset, which will invoke the BIOS and allow it to relock the RAM range.

### 5.12.4 Century Rollover

Chipset detects a rollover when the Year byte (RTC I/O space, index offset 09h) transitions form 99 to 00. Upon detecting the rollover, Chipset sets the NEWCENTURY_STS bit (TCOBASE + 04h, bit 7). If the system is in an S0 state, this causes an SMI#. The SMI# handler can update registers in the RTC RAM that are associated with century value. If the system is in a sleep state (S1–S5) when the century rollover occurs, Chipset also sets the NEWCENTURY_STS bit, but no SMI# is generated. When the system resumes from the sleep state, BIOS should check the NEWCENTURY_STS bit and update the century value in the RTC RAM.

### 5.12.5 Clearing Battery-Backed RTC RAM

Clearing CMOS RAM in an Chipset-based platform can be done by using a jumper on RTCRST# or GPI. Implementations should not attempt to clear CMOS by using a jumper to pull VccRTC low.

#### Using RTCRST# to Clear CMOS

A jumper on RTCRST# can be used to clear CMOS values, as well as reset to default, the state of those configuration bits that reside in the RTC power well. When the RTCRST# is strapped to ground, the RTC_PWR_STS bit (D31:F0:A4h bit 2) will be set and those configuration bits in the RTC power well will be set to their default state. BIOS can monitor the state of this bit, and manually clear the RTC CMOS array once the system is booted. The normal position would cause RTCRST# to be pulled up through a weak pull-up resistor. Table 5-49 shows which bits are set to their default state when RTCRST# is asserted. This RTCRST# jumper technique allows the jumper to be moved and then replaced—all while the system is powered off. Then, once booted, the RTC_PWR_STS can be detected in the set state.

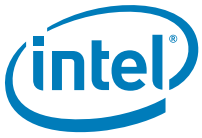


**Table 5-49.Configuration Bits Reset by RTCRST# Assertion (Sheet 1 of 2)**

| Bit Name | Register | Location | Bit(s) | Default State |
|---|---|---|---|---|
| Alarm Interrupt Enable (AIE) | Register B (General Configuration) (RTC_REGB) | I/O space (RTC Index + 0Bh) | 5 | X |
| Alarm Flag (AF) | Register C (Flag Register) (RTC_REGC) | I/O space (RTC Index + 0Ch) | 5 | X |
| SWSMI_RATE_SEL | General PM Configuration 3 Register GEN_PMCON_3 | D31:F0:A4h | 7:6 | 0 |
| SLP_S4# Minimum Assertion Width | General PM Configuration 3 Register GEN_PMCON_3 | D31:F0:A4h | 5:4 | 0 |
| SLP_S4# Assertion Stretch Enable | General PM Configuration 3 Register GEN_PMCON_3 | D31:F0:A4h | 3 | 0 |
| RTC Power Status (RTC_PWR_STS) | General PM Configuration 3 Register GEN_PMCON_3 | D31:F0:A4h | 2 | 0 |
| Power Failure (PWR_FLR) | General PM Configuration 3 Register (GEN_PMCON_3) | D31:F0:A4h | 1 | 0 |
| AFTERG3_EN | General PM Configuration 3 Register GEN_PMCON_3 | D31:F0:A4h | 0 | 0 |
| Power Button Override Status (PRBTNOR_STS) | Power Management 1 Status Register (PM1_STS) | PMBase + 00h | 11 | 0 |
| RTC Event Enable (RTC_EN) | Power Management 1 Enable Register (PM1_EN) | PMBase + 02h | 10 | 0 |
| Sleep Type (SLP_TYP) | Power Management 1 Control (PM1_CNT) | PMBase + 04h | 12:10 | 0 |
| PME_EN | General Purpose Event 0 Enables Register (GPE0_EN) | PMBase + 2Ch | 11 | 0 |
| BATLOW_EN (Netbook Only) | General Purpose Event 0 Enables Register (GPE0_EN) | PMBase + 2Ch | 10 | 0 |

**Table 5-49.Configuration Bits Reset by RTCRST# Assertion (Sheet 2 of 2)**

| Bit Name | Register | Location | Bit(s) | Default State |
|---|---|---|---|---|
| RI_EN | General Purpose Event 0 Enables Register (GPE0_EN) | PMBase + 2Ch | 8 | 0 |
| NEWCENTURY_STS | TCO1 Status Register (TCO1_STS) | TCOBase + 04h | 7 | 0 |
| Intruder Detect (INTRD_DET) | TCO2 Status Register (TCO2_STS) | TCOBase + 06h | 0 | 0 |
| Top Swap (TS) | Backed Up Control Register (BUC) | Chipset Config Registers:Offset 3414h | 0 | X |

**Using a GPI to Clear CMOS**

A jumper on a GPI can also be used to clear CMOS values. BIOS would detect the setting of this GPI on system boot-up, and manually clear the CMOS array.

***Note:*** The GPI strap technique to clear CMOS requires multiple steps to implement. The system is booted with the jumper in new position, then powered back down. The jumper is replaced back to the normal position, then the system is rebooted again.

***Warning:*** Clearing CMOS, using a jumper on VccRTC, must **not** be implemented.

## 5.13 Processor Interface (D31:F0)

Chipset interfaces to the processor with a variety of signals

- Standard Outputs to processor: A20M#, SMI#, NMI, INIT#, INTR, STPCLK#, IGNNE#, CPUSLP# (supported only on Nettop platforms), CPUPWRGD
- Standard Input from processor: FERR#
- Intel SpeedStep® technology output to processor: CPUPWRGOOD (In Netbook configurations)

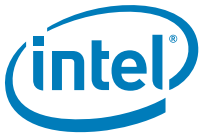
Most Chipset outputs to the processor use standard buffers. Chipset has separate V_CPU_IO signals that are pulled up at the system level to the processor voltage, and thus determines VOH for the outputs to the processor.

### 5.13.1 Processor Interface Signals

This section describes each of the signals that interface between Chipset and the processor(s). Note that the behavior of some signals may vary during processor reset, as the signals are used for frequency strapping.

#### 5.13.1.1 A20M# (Mask A20)

The A20M# signal is active (low) when both of the following conditions are true:



- The ALT_A20_GATE bit (Bit 1 of PORT92 register) is a 0
- The A20GATE input signal is a 0

The A20GATE input signal is expected to be generated by the external microcontroller (KBC).

#### 5.13.1.2 INIT# (Initialization)

The INIT# signal is active (driven low) based on any one of several events described in Table 5-50. When any of these events occur, INIT# is driven low for 16 PCI clocks, then driven high.

***Note:*** The 16-clock counter for INIT# assertion halts while STPCLK# is active. Therefore, if INIT# is supposed to go active while STPCLK# is asserted, it actually goes active after STPCLK# goes inactive.

This section refers to INIT#, but applies to two signals: INIT# and INIT3_3V#, as INIT3_3V# is functionally identical to INIT#, but signaling at 3.3 V.

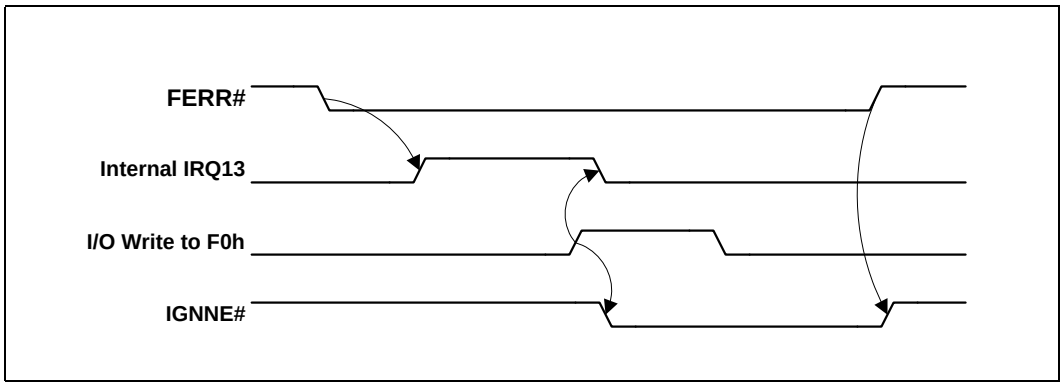
**Table 5-50. INIT# Going Active**

| Cause of INIT# Going Active | Comment |
|---|---|
| Shutdown special cycle from processor. | |
| PORT92 write, where INIT_NOW (bit 0) transitions from a 0 to a 1. | |
| PORTCF9 write, where SYS_RST (bit 1) was a 0 and RST_CPU (bit 2) transitions from 0 to 1. | |
| RCIN# input signal goes low. RCIN# is expected to be driven by the external microcontroller (KBC). | 0 to 1 transition on RCIN# must occur before Chipset will arm INIT# to be generated again.<br>**NOTE:** RCIN# signal is expected to be high during $S3_{HOT}$ and low during $S3_{COLD}$, S4, and S5 states. Transition on the RCIN# signal in those states (or the transition to those states) may not necessarily cause the INIT# signal to be generated to the processor. |
| CPU BIST | To enter BIST, software sets CPU_BIST_EN bit and then does a full processor reset using the CF9 register. |

#### 5.13.1.3 FERR# / IGNNE# (Numeric Coprocessor Error / Ignore Numeric Error)

Chipset supports the coprocessor error function with the FERR#/IGNNE# pins. The function is enabled via the COPROC_ERR_EN bit (Chipset Config Registers:Offset 31FFh:bit 1). FERR# is tied directly to the Coprocessor Error signal of the processor. If FERR# is driven active by the processor, IRQ13 goes active (internally). When it detects a write to the COPROC_ERR register (I/O Register F0h), Chipset negates the internal IRQ13 and drives IGNNE# active. IGNNE# remains active until FERR# is driven inactive. IGNNE# is not driven active unless FERR# is active.

**Figure 5-12. Coprocessor Error Timing Diagram**




If COPROC_ERR_EN is not set, the assertion of FERR# will have not generate an internal IRQ13, nor will the write to F0h generate IGNNE#.

#### 5.13.1.4 NMI (Non-Maskable Interrupt)

Non-Maskable Interrupts (NMIs) can be generated by several sources, as described in Table 5-51.

**Table 5-51.NMI Sources**

| Cause of NMI | Comment |
|---|---|
| SERR# goes active (either internally, externally via SERR# signal, or via message from CPU) | Can instead be routed to generate an SCI, through the NMI2SCI_EN bit (Device 31:Function 0, TCO Base + 08h, bit 11). |
| IOCHK# goes active via SERIRQ# stream (ISA system Error) | Can instead be routed to generate an SCI, through the NMI2SCI_EN bit (Device 31:Function 0, TCO Base + 08h, bit 11). |

#### 5.13.1.5 Stop Clock Request and CPU Sleep (STPCLK# and CPUSLP#)

Chipset power management logic controls these active-low signals. Refer to Section 5.14 for more information on the functionality of these signals.

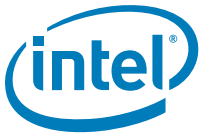
***Note:*** CPU Sleep (CPUSLP#) is supported only on Nettop platforms.

#### 5.13.1.6 CPU Power Good (CPUPWRGOOD)

This signal is connected to the processor's PWRGOOD input. This signal represents a logical AND of chipset's PWROK and VRMPWRGD signals.

#### 5.13.1.7 Deeper Sleep (DPSLP#)

This active-low signal controls the internal gating of the processor's core clock. This signal asserts before and deasserts after the STP_CPU# signal to effectively stop the processor's clock (internally) in the states in which STP_CPU# can be used to stop the processor's clock externally.



### 5.13.2 Dual-Processor Issues (Nettop Only)

#### 5.13.2.1 Signal Differences

In dual-processor designs, some of the processor signals are unused or used differently than for uniprocessor designs.

**Table 5-52.DP Signal Differences**

| Signal | Difference |
|---|---|
| A20M# / A20GATE | Generally not used, but still supported by Chipset. |
| STPCLK# | Used for S1 State as well as preparation for entry to S3–S5<br>Also allows for THERM# based throttling (not via ACPI control methods). Should be connected to both processors. |
| FERR# / IGNNE# | Generally not used, but still supported by Chipset. |

#### 5.13.2.2 Power Management

For Dual-processor configurations in which more than one Stop Grant cycle may be generated, the CPU is expected to count Stop Grant cycles and only pass the last one through to Chipset. This prevents Chipset from getting out of sync with the processor on multiple STPCLK# assertions.

Because the S1 state will have the STPCLK# signal active, the STPCLK# signal can be connected to both processors. However, for ACPI implementations, the BIOS must indicate that Chipset only supports the C1 state for dual-processor designs.

In going to the S1 state for Nettop, multiple Stop-Grant cycles will be generated by the processors. Chipset also has the option to assert the processor's SLP# signal (CPUSLP#). It is assumed that prior to setting the SLP_EN bit (which causes the transition to the S1 state), the processors will not be executing code that is likely to delay the Stop-Grant cycles.

In going to the S3, S4, or S5 states, the system will appear to pass through the S1 state; thus, STPCLK# and SLP# are also used. During the S3, S4, and S5 states, both processors will lose power. Upon exit from those states, the processors will have their power restored.

## 5.14 Power Management (D31:F0)

### 5.14.1 Features

- Support for *Advanced Configuration and Power Interface, Version 2.0 (ACPI)* providing power and thermal management
  - — ACPI 24-Bit Timer
  - — Software initiated throttling of processor performance for Thermal and Power Reduction

  - — Hardware Override to throttle processor performance if system too hot
  - — SCI and SMI# Generation
- PCI PME# signal for Wake Up from Low-Power states
- System Clock Control
  - — (Netbook Only) ACPI C2 state: Stop Grant (using STPCLK# signal) halts processor's instruction stream
  - — (Netbook Only) ACPI C3 State: Ability to halt processor clock (but not memory clock)
  - — (Netbook Only) ACPI C4 State: Ability to lower processor voltage.
  - — (Netbook Only) CLKRUN# Protocol for PCI Clock Starting/Stopping
- System Sleep State Control
  - — ACPI S1 state: Stop Grant (using STPCLK# signal) halts processor's instruction stream (only STPCLK# active, and CPUSLP# optional)
  - — 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
- Streamlined Legacy Power Management for APM-Based Systems

### 5.14.2 Chipset and System Power States

Table 5-53 shows the power states defined for CHIPSET-based platforms. The state names generally match the corresponding ACPI states.

**Table 5-53.General Power States for Systems Using Chipset (Sheet 1 of 2)**

| State / Substates | Legacy Name / Description |
|---|---|
| G0/S0/C0 | **Full On:** Processor operating. Individual devices may be shut down to save power. The different processor operating levels are defined by Cx states, as shown in Table 5-54. Within the C0 state, Chipset can throttle the processor using the STPCLK# signal to reduce power consumption. The throttling can be initiated by software or by the operating system or BIOS. |
| G0/S0/C1 | **Auto-Halt:** Processor has executed an AutoHalt instruction and is not executing code. The processor snoops the bus and maintains cache coherency. |
| G0/S0/C2 (Netbook Only) | **Stop-Grant:** The STPCLK# signal goes active to the processor. The processor performs a Stop-Grant cycle, halts its instruction stream, and remains in that state until the STPCLK# signal goes inactive. In the Stop-Grant state, the processor snoops the bus and maintains cache coherency. |
| G0/S0/C3 (Netbook Only) | **Stop-Clock:** The STPCLK# signal goes active to the processor. The processor performs a Stop-Grant cycle, halts its instruction stream. Chipset then asserts DPSLP# followed by STP_CPU#, which forces the clock generator to stop the processor clock. This is also used for Intel SpeedStep® technology support. Accesses to memory (by graphics, PCI, or internal units) is not permitted while in a C3 state. |

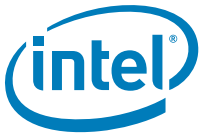


**Table 5-53.General Power States for Systems Using Chipset (Sheet 2 of 2)**

| State/ Substates | Legacy Name / Description |
|---|---|
| G0/S0/C4 (Netbook Only) | **Stop-Clock with Lower Processor Voltage:** This closely resembles the G0/S0/C3 state. However, after Chipset has asserted STP_CPU#, it then lowers the voltage to the processor. This reduces the leakage on the processor. Prior to exiting the C4 state, Chipset increases the voltage to the processor. |
| G1/S1 | **Stop-Grant:** Similar to G0/S0/C2 state. Chipset also has the option to assert the CPUSLP# signal to further reduce processor power consumption (Nettop Only).<br>Note: The behavior for this state is slightly different when supporting iA64 processors. |
| G1/S3 | **Suspend-To-RAM (STR):** The system context is maintained in system DRAM, but power is shut off to non-critical circuits. Memory is retained, and refreshes continue. All clocks stop except RTC clock. |
| G1/S4 | **Suspend-To-Disk (STD):** The context of the system is maintained on the disk. All power is then shut off to the system except for the logic required to resume. |
| G2/S5 | **Soft Off (SOFF):** System context is not maintained. All power is shut off except for the logic required to restart. A full boot is required when waking. |
| G3 | **Mechanical OFF (MOFF):** System context not maintained. All power is shut off 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, transition will depends on the state just prior to the entry to G3 and the AFTERG3 bit in the GEN_PMCON3 register (D31:F0, offset A4). Refer to Table 5-61 for more details. |

Table 5-54 shows the transitions rules among the various states. Note that transitions among the various states may appear to temporarily transition through intermediate states. For example, in going from S0 to S1, it may appear to pass through the G0/S0/C2 states. These intermediate transitions and states are not listed in the table.

**Table 5-54.State Transition Rules for Chipset (Sheet 1 of 2)**

| Present State | Transition Trigger | Next State |
|---|---|---|
| G0/S0/C0 | • Processor halt instruction<br>• Level 2 Read<br>• Level 3 Read (Netbook Only)<br>• Level 4 Read (Netbook Only)<br>• SLP_EN bit set<br>• Power Button Override<br>• Mechanical Off/Power Failure | • G0/S0/C1<br>• G0/S0/C2<br>• G0/S0/C2, G0/S0/C3 or G0/S0/C4 - depending on C4onC3_EN bit (D31:F0:Offset A0h:bit 7) and BM_STS_ZERO_EN bit (D31:F0:Offset A9h:bit 2) (Netbook Only)<br>• G1/Sx or G2/S5 state<br>• G2/S5<br>• G3 |
| G0/S0/C1 | • Any Enabled Break Event<br>• STPCLK# goes active<br>• Power Button Override<br>• Power Failure | • G0/S0/C0<br>• G0/S0/C2<br>• G2/S5<br>• G3 |

**Table 5-54.State Transition Rules for Chipset (Sheet 2 of 2)**

| Present State | Transition Trigger | Next State |
|---|---|---|
| G0/S0/C2 (Netbook Only) | • Any Enabled Break Event<br>• Power Button Override<br>• Power Failure<br>• Previously in C3/C4 and bus masters idle | • G0/S0/C0<br>• G2/S5<br>• G3<br>• C3 or C4 - depending on PDME bit (D31:F0: Offset A9h: bit 4) |
| G0/S0/C3 (Netbook Only) | • Any Enabled Break Event<br>• Any Bus Master Event<br>• Power Button Override<br>• Power Failure<br>• Previously in C4 and bus masters idle | • G0/S0/C0<br>• G0/S0/C2 - if PUME bit (D31:F0: Offset A9h: bit 3) is set, else G0/S0/C0<br>• G2/S5<br>• G3<br>• C4 - depending on PDME bit (D31:F0: Offset A9h: bit 4 |
| G0/S0/C4 (Netbook Only) | • Any Enabled Break Event<br>• Any Bus Master Event<br>• Power Button Override<br>• Power Failure | • G0/S0/C0<br>• G0/S0/C2 - if PUME bit (D31:F0: Offset A9h: bit 3) is set, else G0/S0/C0<br>• G2/S5<br>• G3 |
| G1/S1, G1/S3, or G1/S4 | • Any Enabled Wake Event<br>• Power Button Override<br>• Power Failure | • G0/S0/C0 (See Note 2)<br>• G2/S5<br>• G3 |
| G2/S5 | • Any Enabled Wake Event<br>• Power Failure | • G0/S0/C0 (See Note 2)<br>• G3 |
| G3 | • Power Returns | • Optional to go to S0/C0 (reboot) or G2/S5 (stay off until power button pressed or other wake event). (See Note 1 and 2) |

**NOTES:**
1. Some wake events can be preserved through power failure.
2. Transitions from the S1–S5 or G3 states to the S0 state are deferred until BATLOW# is inactive in Netbook configurations.

## 5.14.3 System Power Planes

The system has several independent power planes, as described in Table 5-55. Note that when a particular power plane is shut off, it should go to a 0 V level.

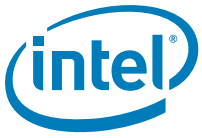


**Table 5-55.System Power Plane**

| Plane | Controlled By | Description |
|---|---|---|
| CPU | SLP_S3# signal | The SLP_S3# signal can be used to cut the power to the processor completely. For Netbook systems, the DPRSLPVR support allows lowering the processor's voltage during the C4 state.<br>$S3_{HOT}$: The new $S3_{HOT}$ state keeps more of the platform logic, including Chipset core well, powered to reduce the cost of external power plane logic. SLP_S3# is only used to remove power to the processor and to shut system clocks. This impacts the board design, but there is no specific Chipset bit or strap needed to indicate which option is selected. |
| MAIN | SLP_S3# signal ($S3_{COLD}$) or SLP_S4# signal ($S3_{HOT}$) | $S3_{COLD}$: When SLP_S3# goes active, power can be shut off to any circuit not required to wake the system from the S3 state. Since the S3 state requires that the memory context be preserved, power must be retained to the main memory.<br>The processor, devices on the PCI bus, LPC I/F, and graphics will typically be shut off when the Main power plane is shut, although there may be small subsections powered.<br>$S3_{HOT}$: SLP_S4# is used to cut the main power well, rather than using SLP_S3#. This impacts the board design, but there is no specific Chipset bit or strap needed to indicate which option is selected. |
| MEMORY | SLP_S4# signal SLP_S5# signal | When the SLP_S4# goes active, power can be shut off to any circuit not required to wake the system from the S4. Since the memory context does not need to be preserved in the S4 state, the power to the memory can also be shut down.<br>When SLP_S5# goes active, power can be shut 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. |
| DEVICE[n] | GPIO | Individual subsystems may have their own power plane. For example, GPIO signals may be used to control the power to disk drives, audio amplifiers, or the display screen. |

## 5.14.4 SMI#/SCI Generation

On any SMI# event taking place, Chipset asserts SMI# to the processor, which causes it to enter SMM space. SMI# remains active until the EOS bit is set. When the EOS bit is set, SMI# goes inactive for a minimum of 4 PCICLK. If another SMI event occurs, SMI# is driven active again.

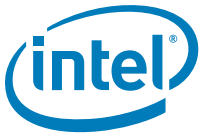
The SCI is a level-mode interrupt that is typically handled by an ACPI-aware operating system. In non-APIC systems (which is the default), the SCI IRQ is routed to one of the 8259 interrupts (IRQ 9, 10, or 11). The 8259 interrupt controller must be programmed to level mode for that interrupt.

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 (see Section 13.1.14). The interrupt remains asserted until all SCI sources are removed.

Table 5-56 shows which events can cause an SMI# and SCI. Note that 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 5-56.Causes of SMI# and SCI (Sheet 1 of 2)**

| Cause[1-5] | SCI | SMI | Additional Enables | Where Reported |
|---|---|---|---|---|
| PME# | Yes | Yes | PME_EN=1 | PME_STS |
| PME_B0 (internal EHCI controller) | Yes | Yes | PME_B0_EN=1 | PME_B0_STS |
| PCI Express* PME Messages | Yes | Yes | PCI_EXP_EN=1<br>(Not enabled for SMI) | PCI_EXP_STS |
| PCI Express Hot Plug Message | Yes | Yes | HOT_PLUG_EN=1<br>(Not enabled for SMI) | HOT_PLUG_STS |
| Power Button Press | Yes | Yes | PWRBTN_EN=1 | PWRBTN_STS |
| Power Button Override[6]) | Yes | No | None | PRBTNOR_STS |
| RTC Alarm | Yes | Yes | RTC_EN=1 | RTC_STS |
| Ring Indicate | Yes | Yes | RI_EN=1 | RI_STS |
| USB#1 wakes | Yes | Yes | USB1_EN=1 | USB1_STS |
| USB#2 wakes | Yes | Yes | USB2_EN=1 | USB2_STS |
| USB#3 wakes | Yes | Yes | USB3_EN=1 | USB3_STS |
| USB#4 wakes | Yes | Yes | USB4_EN=1 | USB4_STS |
| THRM# pin active | Yes | Yes | THRM_EN=1 | THRM_STS |
| ACPI Timer overflow (2.34 sec.) | Yes | Yes | TMROF_EN=1 | TMROF_STS |
| Any GPI[7] | Yes | Yes | GPI[x]_Route=10 (SCI)<br>GPI[x]_Route=01 (SMI)<br>GPE0[x]_EN=1 | GPI[x]_STS<br>GPE0_STS |
| TCO SCI Logic | Yes | No | TCOSCI_EN=1 | TCOSCI_STS |
| TCO SCI message from CPU | Yes | No | none | MCHSCI_STS |
| TCO SMI Logic | No | Yes | TCO_EN=1 | TCO_STS |
| TCO SMI — Year 2000 Rollover | No | Yes | none | NEWCENTURY_STS |
| TCO SMI — TCO TIMEROUT | No | Yes | none | TIMEOUT |
| TCO SMI — OS writes to TCO_DAT_IN register | No | Yes | none | OS_TCO_SMI |
| TCO SMI — Message from CPU | No | Yes | none | MCHSMI_STS |
| TCO SMI — NMI occurred (and NMIs mapped to SMI) | No | Yes | NMI2SMI_EN=1 | NMI2SMI_STS |
| TCO SMI — INTRUDER# signal goes active | No | Yes | INTRD_SEL=10 | INTRD_DET |



### Table 5-56.Causes of SMI# and SCI (Sheet 2 of 2)

| Cause[1-5] | SCI | SMI | Additional Enables | Where Reported |
|---|---|---|---|---|
| TCO SMI — Change of the BIOSWP bit from 0 to 1 | No | Yes | BLD=1 | BIOSWR_STS |
| TCO SMI — Write attempted to BIOS | No | Yes | BIOSWP=1 | BIOSWR_STS |
| BIOS_RLS written to | 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 |
| Enhanced USB Intel Specific Event | No | Yes | INTEL_USB2_EN = 1 | INTEL_USB2_STS |
| UHCI USB Legacy logic | No | Yes | LEGACY_USB_EN=1 | LEGACY_USB_STS |
| Serial IRQ SMI reported | No | Yes | none | SERIRQ_SMI_STS |
| Device monitors match address in its range | No | Yes | none | DEVMON_STS, DEVACT_STS |
| SMBus Host Controller | No | Yes | SMB_SMI_EN Host Controller Enabled | SMBus host status reg. |
| SMBus Slave SMI message | No | Yes | none | SMBUS_SMI_STS |
| SMBus SMBALERT# signal active | No | Yes | none | SMBUS_SMI_STS |
| SMBus Host Notify message received | No | Yes | HOST_NOTIFY_INTREN | SMBUS_SMI_STS HOST_NOTIFY_STS |
| (Netbook) 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 |

**NOTES:**
1. SCI_EN must be 1 to enable SCI. 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. Chipset must have SMI# fully enabled when Chipset 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. Only GPI[15:0] may generate an SMI# or SCI.

#### 5.14.4.1 PCI Express* SCI

PCI Express ports and the CPU (via DMI) have the ability to cause PME using messages. When a PME message is received, Chipset will set the PCI_EXP_STS bit. If the PCI_EXP_EN bit is also set, Chipset can cause an SCI via the GPE1_STS register.

#### 5.14.4.2 PCI Express* Hot-Plug

PCI Express has a Hot-Plug mechanism and is capable of generating a SCI via the GPE1 register. It is also capable of generating an SMI. However, it is not capable of generating a wake event.

### 5.14.5 Dynamic Processor Clock Control

Chipset has extensive control for dynamically starting and stopping system clocks. The clock control is used for transitions among the various S0/Cx states, and processor throttling. Each dynamic clock control method is described in this section. The various sleep states may also perform types of non-dynamic clock control.

Chipset supports the ACPI C0 and C1 states (in Nettop) or C0, C1, C2, C3 and C4 (in Netbook) states.

The Dynamic Processor Clock control is handled using the following signals:

- STPCLK#: Used to halt processor instruction stream.
- (Netbook) STP_CPU#: Used to stop processor's clock
- (Netbook) DPSLP#: Used to force Deeper Sleep for processor.
- (Netbook) DPRSLPVR: Used to lower voltage of VRM during C4 state.
- (Netbook) DPRSTP#: Used to lower voltage of VRM during C4 state

The C1 state is entered based on the processor performing an auto halt instruction.

(Netbook) The C2 state is entered based on the processor reading the Level 2 register in Chipset. It can also be entered from C3 or C4 states if bus masters require snoops and the PUME bit (D31:F0: Offset A9h: bit 3) is set.

(Netbook) The C3 state is entered based on the processor reading the Level 3 register in Chipset and when the C4 on C3_EN bit is clear (D31:F0:Offset A0:bit 7). This state can also be entered after a temporary return to C2 from a prior C3 or C4 state.

(Netbook) The C4 state is entered based on the processor reading the Level 4 register in Chipset, or by reading the Level 3 register when the C4onC3_EN bit is set. This state can also be entered after a temporary return to C2 from a prior C4 state.

A C1 state in Nettop Only or a C1, C2, C3, or C4 state in Netbook Only ends due to a Break event. Based on the break event, Chipset returns the system to C0 state.

(Netbook) Table 5-57 lists the possible break events from C2, C3, or C4. The break events from C1 are indicated in the processor's datasheet.

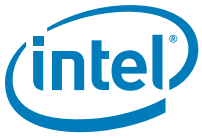


**Table 5-57.Break Events (Netbook)**

| Event | Breaks from | Comment |
|---|---|---|
| Any unmasked interrupt goes active | C2, C3, C4 | IRQ[0:15] when using the 8259s, IRQ[0:23] for I/O APIC. Since SCI is an interrupt, any SCI will also be a break event. |
| Any internal event that cause an NMI or SMI# | C2, C3, C4 | Many possible sources |
| Any internal event that cause INIT# to go active | C2, C3, C4 | Could be indicated by the keyboard controller via the RCIN input signal. |
| Any bus master request (internal, external or DMA, or BM_BUSY#) goes active and BM_RLD=1 (D31:F0:Offset PMBASE+04h: bit 1) | C3, C4 | Need to wake up processor so it can do snoops<br><br>**NOTE:** If the PUME bit (D31:F0: Offset A9h: bit 3) is set, then bus master activity will NOT be treated as a break event. Instead, there will be a return only to the C2 state. |
| Processor Pending Break Event Indication | C2, C3, C4 | Only available if FERR# enabled for break event indication (See FERR# Mux Enable in GCS, Chipset Config Registers:Offset 3410h:bit 6) |

### 5.14.5.1 Transition Rules among S0/Cx and Throttling States

The following priority rules and assumptions apply among the various S0/Cx and throttling states:

- Entry to any S0/Cx state is mutually exclusive with entry to any S1–S5 state. This is because the processor can only perform one register access at a time and Sleep states have higher priority than thermal throttling.
- When the SLP_EN bit is set (system going to a S1–S5 sleep state), the THTL_EN and FORCE_THTL bits can be internally treated as being disabled (no throttling while going to sleep state).
- (Netbook) If the THTL_EN or FORCE_THTL bits are set, and a Level 2, Level 3, or Level 4 read then occurs, the system should immediately go and stay in a C2, C3, or C4 state until a break event occurs. A Level 2, Level 3, or Level 4 read has higher priority than the software initiated throttling.
- (Netbook) After an exit from a C2, C3, or C4 state (due to a Break event), and if the THTL_EN or FORCE_THTL bits are still set the system will continue to throttle STPCLK#. Depending on the time of break event, the first transition on STPCLK# active can be delayed by up to one THRM period (1024 PCI clocks = 30.72 µs).
- The Host controller must post Stop-Grant cycles in such a way that the processor gets an indication of the end of the special cycle prior to Chipset observing the Stop-Grant cycle. This ensures that the STPCLK# signals stays active for a sufficient period after the processor observes the response phase.

- (Netbook) If in the C1 state and the STPCLK# signal goes active, the processor will generate a Stop-Grant cycle, and the system should go to the C2 state. When STPCLK# goes inactive, it should return to the C1 state.

#### 5.14.5.2 Deferred C3/C4 (Netbook Only)

Due to the new DMI protocol, if there is any bus master activity (other than true isoch), then the C0-to-C3 transition will pause at the C2 state. Chipset will keep the processor in a C2 state until:

- Chipset does not detect bus master activity.
- A break event occurs. In this case, Chipset will perform the C2 to C0 sequence. Note that bus master traffic is not a break event in this case.

To take advantage of the Deferred C3/C4 mode, the BM_STS_ZERO_EN bit must be set. This will cause the BM_STS bit to read as 0 even if some bus master activity is present. If this is not done, then the software may avoid even attempting to go to the C3 or C4 state if it sees the BM_STS bit as 1.

If the PUME bit (D31:F0: Offset A9h: bit 3) is 0, then Chipset will treat bus master activity as a break event. When reaching the C2 state, if there is any bus master activity, Chipset will return the processor to a C0 state.

#### 5.14.5.3 POPUP (Auto C3/C4 to C2) (Netbook Only)

When the PUME bit (D31:F0: Offset A9h: bit 3) is set, Chipset enables a mode of operation where standard (non-isoch) bus master activity will not be treated as a full break event from the C3 or C4 states. Instead, these will be treated merely as bus master events and return the platform to a C2 state, and thus allow snoops to be performed.

After returning to the C2 state, the bus master cycles will be sent to the CPU, even if the ARB_DIS bit is set.

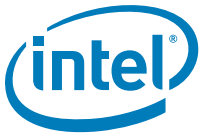
#### 5.14.5.4 POPDOWN (Auto C2 to C3/C4) (Netbook Only)

After returning to the C2 state from C3/C4, it the PDME bit (D31:F0: Offset A9h: bit 4) is set, the platform can return to a C3 or C4 state (depending on where it was prior to going back up to C2). This behaves similar to the Deferred C3/C4 transition, and will keep the processor in a C2 state until:

- Bus masters are no longer active.
- A break event occurs. Note that bus master traffic is not a break event in this case.

### 5.14.6 Dynamic PCI Clock Control (Netbook Only)

The PCI clock can be dynamically controlled independent of any other low-power state. This control is accomplished using the CLKRUN# protocol as described in the *PCI Mobile Design Guide,* and is transparent to software.



The Dynamic PCI Clock control is handled using the following signals:

- CLKRUN#: Used by PCI and LPC peripherals to request the system PCI clock to run
- STP_PCI#: Used to stop the system PCI clock

***Note:*** The 33 MHz clock to Chipset is “free-running” and is not affected by the STP_PCI# signal.

#### 5.14.6.1 Conditions for Checking the PCI Clock

When there is a lack of PCI activity Chipset has the capability to stop the PCI clocks to conserve power. “PCI activity” is defined as any activity that would require the PCI clock to be running.

Any of the following conditions will indicate that it is **not okay** to stop the PCI clock:

- Cycles on PCI or LPC
- Cycles of any internal device that would need to go on the PCI bus
- SERIRQ activity

**Behavioral Description**

- When there is a lack of activity (as defined above) for 29 PCI clocks, Chipset deasserts (drive high) CLKRUN# for 1 clock and then tri-states the signal.

#### 5.14.6.2 Conditions for Maintaining the PCI Clock

PCI masters or LPC devices that wish to maintain the PCI clock running will observe the CLKRUN# signal deasserted, and then must re-assert if (drive it low) within 3 clocks.

- When Chipset has tri-stated the CLKRUN# signal after deasserting it, Chipset then checks to see if the signal has been re-asserted (externally).
- After observing the CLKRUN# signal asserted for 1 clock, Chipset again starts asserting the signal.
- If an internal device needs the PCI bus, Chipset asserts the CLKRUN# signal.

#### 5.14.6.3 Conditions for Stopping the PCI Clock

- If no device re-asserts CLKRUN# once it has been deasserted for at least 6 clocks, Chipset stops the PCI clock by asserting the STP_PCI# signal to the clock synthesizer.

#### 5.14.6.4 Conditions for Re-Starting the PCI Clock

- A peripheral asserts CLKRUN# to indicate that it needs the PCI clock re-started.
- When Chipset observes the CLKRUN# signal asserted for 1 (free running) clock, Chipset deasserts the STP_PCI# signal to the clock synthesizer within 4 (free running) clocks.
- Observing the CLKRUN# signal asserted externally for 1 (free running) clock, Chipset again starts driving CLKRUN# asserted.

If an internal source requests the clock to be re-started, Chipset re-asserts CLKRUN#, and simultaneously deasserts the STP_PCI# signal.

#### 5.14.6.5 LPC Devices and CLKRUN# (Netbook Only)

If an LPC device (of any type) needs the 33 MHz PCI clock, such as for LPC DMA (Netbook Only) or LPC serial interrupt, then it can assert CLKRUN#. Note that LPC devices running DMA or bus master cycles will not need to assert CLKRUN#, since Chipset asserts it on their behalf.

The LDRQ# inputs are ignored by Chipset when the PCI clock is stopped to the LPC devices in order to avoid misinterpreting the request. Chipset assumes that only one more rising PCI clock edge occurs at the LPC device after the assertion of STP_PCI#. Upon deassertion of STP_PCI#, Chipset assumes that the LPC device receives its first clock rising edge corresponding to chipset's second PCI clock rising edge after the deassertion.

### 5.14.7 Sleep States

#### 5.14.7.1 Sleep State Overview

Chipset directly supports different sleep states (S1–S5) that are entered by setting the SLP_EN bit, or due to a Power Button press. The entry to the Sleep states are based on several assumptions:

- Entry to a Cx state is mutually exclusive with entry to a Sleep state. This is because the processor can only perform one register access at a time. A request to Sleep has higher priority than throttling.
- Prior to setting the SLP_EN bit, the software turns off processor-controlled throttling. Note that thermal throttling cannot be disabled, but setting the SLP_EN bit disables thermal throttling (since S1–S5 sleep state has higher priority).
- The G3 state cannot be entered via any software mechanism. The G3 state indicates a complete loss of power.

#### 5.14.7.2 Initiating Sleep State

Sleep states (S1–S5) are initiated by:

- Masking interrupts, turning off all bus master enable bits, setting the desired type in the SLP_TYP field, and then setting the SLP_EN bit. The hardware then attempts to gracefully put the system into the corresponding Sleep state.
- Pressing the PWRBTN# Signal for more than 4 seconds to cause a Power Button Override event. In this case the transition to the S5 state is less graceful, since there are no dependencies on observing Stop-Grant cycles from the processor or on clocks other than the RTC clock.

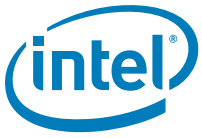


**Table 5-58.Sleep Types**

| Sleep Type | Comment |
|---|---|
| S1 | Chipset asserts the STPCLK# signal. It also has the option to assert CPUSLP# signal (only supported on Nettop platforms). This lowers the processor's power consumption. No snooping is possible in this state. |
| S3 | Chipset 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 | Chipset asserts SLP_S3# and SLP_S4#. The SLP_S4# signal shuts off the power to the memory subsystem. Only devices needed to wake from this state should be powered. |
| S5 | Same power state as S4. Chipset asserts SLP_S3#, SLP_S4# and SLP_S5#. |

### 5.14.7.3 Exiting Sleep States

Sleep states (S1–S5) are exited based on Wake events. The Wake events forces the system to a full on state (S0), although some non-critical subsystems might still be shut off and have to be brought back manually. For example, the hard disk may be shut off during a sleep state, and have to be enabled via a GPIO pin before it can be used.

Upon exit from Chipset-controlled Sleep states, the WAK_STS bit is set. The possible causes of Wake Events (and their restrictions) are shown in Table 5-59.

***Note:*** (Netbook Only) If the BATLOW# signal is asserted, Chipset does not attempt to wake from an S1–S5 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 Chipset, and the system wakes after BATLOW# is de-asserted.

**Table 5-59.Causes of Wake Events**

| Cause | States Can Wake From[1] | How Enabled |
|---|---|---|
| RTC Alarm | S1–S5[2] | Set RTC_EN bit in PM1_EN register |
| Power Button | S1–S5 | Always enabled as Wake event |
| GPI[0:15] | S1–S5[2] | GPE0_EN register<br><br>**NOTE:** GPIs that are in the core well are not capable of waking the system from sleep states where the core well is not powered. |
| Classic USB | S1–S5 | Set USB1_EN, USB 2_EN, USB3_EN, and USB4_EN bits in GPE0_EN register |
| LAN | S1–S5 | Will use PME#. Wake enable set with LAN logic. |
| RI# | S1–S5[2] | Set RI_EN bit in GPE0_EN register |
| Intel HD Audio | S1–S5[2] | Set AC97_EN bit in GPE0_EN register |
| Primary PME# | S1–S5 | PME_B0_EN bit in GPE0_EN register |
| Secondary PME# | S1–S5 | Set PME_EN bit in GPE0_EN register. |

**Table 5-59.Causes of Wake Events**

| Cause | States Can Wake From[1] | How Enabled |
|---|---|---|
| PCI_EXP_WAKE# | S1–S5 | PCI_EXP_WAKE bit[3] |
| PCI_EXP PME Message | S1 | Must use the PCI Express* WAKE# pin rather than messages for wake from S3,S4, or S5. |
| SMBALERT# | S1–S5 | Always enabled as Wake event |
| SMBus Slave Message | S1–S5 | Wake/SMI# command always enabled as a Wake event. Note: SMBus Slave Message can wake the system from S1–S5, as well as from S5 due to Power Button Override. |
| SMBus Host Notify message received | S1–S5 | HOST_NOTIFY_WKEN bit SMBus Slave Command register. Reported in the SMB_WAK_STS bit in the GPEO_STS register. |

**NOTES:**
1. If in the S5 state due to a powerbutton override or THRMTRIP#, the possible wake events are due to Power Button, Hard Reset Without Cycling (See Command Type 3 in Table 5-81), and Hard Reset System (See Command Type 4 in Table 5-81).
2. This is a wake event from S5 only if the sleep state was entered by setting the SLP_EN and SLP_TYP bits via software, or if there is a power failure.
3. When the WAKE# pin is active and the PCI Express device is enabled to wake the system, Chipset will wake the platform.

It is important to understand that the various GPIs have different levels of functionality when used as wake events. The GPIs that reside in the core power well can only generate wake events from sleep states where the core well is powered. Also, only certain GPIs are “ACPI Compliant,” meaning that their Status and Enable bits reside in ACPI I/O space. Table 5-60 summarizes the use of GPIs as wake events.

**Table 5-60.GPI Wake Events**

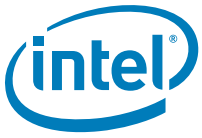
| GPI | Power Well | Wake From | Notes |
|---|---|---|---|
| GPI[12, 7:0] | Core | S1 | ACPI Compliant |
| GPI[15:13,11:8] | Resume | S1–S5 | ACPI Compliant |

The latency to exit the various Sleep states varies greatly and is heavily dependent on power supply design, so much so that the exit latencies due to Chipset are insignificant.

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

PCI Express ports can wake the platform from any sleep state (S1, S3, S4, or S5) using the WAKE# pin. WAKE# is treated as a wake event, but does not cause any bits to go active in the GPE_STS register.

PCI Express ports and the CPU (via DMI) have the ability to cause PME using messages. When a PME message is received, Chipset will set the PCI_EXP_STS bit.



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

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

The AFTER_G3 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 RSMRST# is low (G3 state), the PWRBTN_STS bit is reset. When Chipset exits G3 after power returns (RSMRST# goes high), the PWRBTN# signal is already high (because $V_{CC}$-standby goes high before RSMRST# goes high) and the PWRBTN_STS bit is 0.
2. **RI#:** RI# does not have an internal pull-up. Therefore, if this signal is enabled as a wake event, it is important to keep this signal powered during the power loss event. If this signal goes low (active), when power returns the RI_STS bit is set and the system interprets that as a wake event.
3. **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 RSMRST# goes low.

Chipset monitors both PWROK and RSMRST# to detect for power failures. If PWROK goes low, the PWROK_FLR bit is set. If RSMRST# goes low, PWR_FLR is set.

***Note:*** 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 5-61.Transitions Due to Power Failure**

| State at Power Failure | AFTERG3_EN bit | Transition When Power Returns |
|---|---|---|
| S0, S1, S3 | 1<br>0 | S5<br>S0 |
| S4 | 1<br>0 | S4<br>S0 |
| S5 | 1<br>0 | S5<br>S0 |

### 5.14.8 Thermal Management

Chipset has mechanisms to assist with managing thermal problems in the system.

#### 5.14.8.1 THRM# Signal

The THRM# signal is used as a status input for a thermal sensor. Based on the THRM# signal going active, Chipset generates an SMI# or SCI (depending on SCI_EN).

If the THRM_POL bit is set low, when the THRM# signal goes low, the THRM_STS bit will be set. This is an indicator that the thermal threshold has been exceeded. If the THRM_EN bit is set, then when THRM_STS goes active, either an SMI# or SCI will be generated (depending on the SCI_EN bit being set).

The power management software (BIOS or ACPI) can then take measures to start reducing the temperature. Examples include shutting off unwanted subsystems, or halting the processor.

By setting the THRM_POL bit to high, another SMI# or SCI can optionally be generated when the THRM# signal goes back high. This allows the software (BIOS or ACPI) to turn off the cooling methods.

***Note:*** THRM# assertion does not cause a TCO event message in S3 or S4. The level of the signal is not reported in the heartbeat message.

#### 5.14.8.2 Processor Initiated Passive Cooling

This mode is initiated by software setting the THTL_EN (PMBASE+10h:bit 4) or FORCE_THTL (PMBASE+10h:bit 8) bits.

Software sets the THTL_DTY (PMBASE+10h:bits 3:1) or THRM_DTY (PMBASE+10h:bits 7:5) bits to select throttle ratio and THTL_EN or FORCE_THTL bits to enable the throttling.

Throttling results in STPCLK# active for a minimum time of 12.5% and a maximum of 87.5%. The period is 1024 PCI clocks. Thus, the STPCLK# signal can be active for as little as 128 PCI clocks or as much as 896 PCI clocks. The actual slowdown (and cooling) of the processor depends on the instruction stream, because the processor is allowed to finish the current instruction. Furthermore, Chipset waits for the STOP-GRANT cycle before starting the count of the time the STPCLK# signal is active.

#### 5.14.8.3 THRM# Override Software Bit

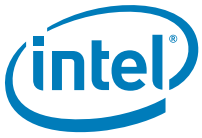
The FORCE_THTL bit allows the BIOS to force passive cooling, independent of the ACPI software (which uses the THTL_EN and THTL_DTY bits). If this bit is set, Chipset starts throttling using the ratio in the THRM_DTY field.

When this bit is cleared Chipset stops throttling, unless the THTL_EN bit is set (indicating that ACPI software is attempting throttling).

If both the THTL_EN and FORCE_THTL bits are set, then Chipset should use the duty cycle defined by the THRM_DTY field, not the THTL_DTY field.

#### 5.14.8.4 Active Cooling

Active cooling involves fans. The GPIO signals from Chipset can be used to turn on/off a fan.



## 5.14.9 Event Input Signals and Their Usage

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

### 5.14.9.1 PWRBTN# (Power Button)

Chipset PWRBTN# signal operates as a “Fixed Power Button” as described in the *Advanced Configuration and Power Interface, Version 2.0b.* PWRBTN# signal has a 16 ms de-bounce on the input. The state transition descriptions are included in Table 5-62. Note that the transitions start as soon as the PWRBTN# is pressed (but after the debounce logic), and does not depend on when the Power Button is released.

***Note:*** During the time that the SLP_S4# signal is stretched for the minimum assertion width (if enabled), the Power Button is not a wake event. Refer to Power Button Override Function section below for further detail.

**Table 5-62.Transitions Due to Power Button**

| Present State | Event | Transition / Action | Comment |
|---|---|---|---|
| S0/Cx | PWRBTN# goes low | SMI# or SCI generated (depending on SCI_EN) | Software typically initiates a Sleep state |
| S1–S5 | PWRBTN# goes low | Wake Event. Transitions to S0 state | Standard wakeup |
| G3 | PWRBTN# pressed | None | No effect since no power<br>Not latched nor detected |
| S0–S4 | PWRBTN# held low for at least 4 consecutive seconds | Unconditional transition to S5 state | No dependence on processor (e.g., Stop-Grant cycles) or any other subsystem |

#### Power Button Override Function

If PWRBTN# is observed active for at least four consecutive seconds, the state machine should unconditionally transition to the G2/S5 state, regardless of present state (S0–S4), even if PWROK is not active. In this case, the transition to the G2/S5 state should not depend on any particular response from the processor (e.g., a Stop-Grant cycle), nor any similar dependency from any other subsystem.

The PWRBTN# status is readable to check if the button is currently being pressed or has been released. The status is taken after the de-bounce, and is readable via the PWRBTN_LVL bit.

***Note:*** The 4-second PWRBTN# assertion should only be used if a system lock-up has occurred. The 4-second timer starts counting when Chipset is in a S0 state. If the PWRBTN# signal is asserted and held active when the system is in a suspend state (S1–S5), the assertion causes a wake event. Once the system has resumed to the S0 state, the 4-second timer starts.

***Note:*** During the time that the SLP_S4# signal is stretched for the minimum assertion width (if enabled by D31:F0:A4h bit 3), the Power Button is not a wake event. As a result, it is conceivable that the user will press and continue to hold the Power Button waiting for the system to awake. Since a 4-second press of the Power Button is already defined as an Unconditional Power down, the power button timer will be forced to inactive while the power-cycle timer is in progress. Once the power-cycle timer has expired, the Power Button awakes the system. Once the minimum SLP_S4# power cycle expires, the Power Button must be pressed for another 4 to 5 seconds to create the Override condition to S5.

**Sleep Button**

The *Advanced Configuration and Power Interface, Version 2.0b* defines an optional Sleep button. It differs from the power button in that it only is a request to go from S0 to S1–S4 (not S5). Also, in an S5 state, the Power Button can wake the system, but the Sleep Button cannot.

Although Chipset does not include a specific signal designated as a Sleep Button, one of the GPIO signals can be used to create a “Control Method” Sleep Button. See the *Advanced Configuration and Power Interface, Version 2.0b* for implementation details.

### 5.14.9.2 RI# (Ring Indicator)

The Ring Indicator can cause a wake event (if enabled) from the S1–S5 states. Table 5-63 shows when the wake event is generated or ignored in different states. If in the G0/S0/Cx states, Chipset generates an interrupt based on RI# active, and the interrupt will be set up as a Break event.

**Table 5-63.Transitions Due to RI# Signal**

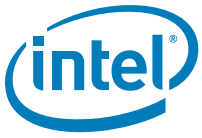
| Present State | Event | RI_EN | Event |
|---|---|---|---|
| S0 | RI# Active | X | Ignored |
| S1–S5 | RI# Active | 0<br>1 | Ignored<br>Wake Event |

***Note:*** Filtering/Debounce on RI# will not be done in CHIPSET. It can be in modem or external.

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

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

There is also an internal PME_B0 bit. This is separate from the external PME# signal and can cause the same effect.



#### 5.14.9.4 SYS_RESET# Signal

When the SYS_RESET# pin is detected as active after the 16 ms debounce logic, Chipset attempts to perform a “graceful” reset, by waiting up to 25 ms for the SMBus to go idle. If the SMBus is idle when the pin is detected active, the reset occurs immediately; otherwise, the counter starts. If at any point during the count the SMBus goes idle the reset occurs. If, however, the counter expires and the SMBus is still active, a reset is forced upon the system even though activity is still occurring.

Once the reset is asserted, it remains asserted for 5 to 6 ms regardless of whether the SYS_RESET# input remains asserted or not. It cannot occur again until SYS_RESET# has been detected inactive after the debounce logic, and the system is back to a full S0 state with PLTRST# inactive. Note that if bit 3 of the CF9h I/O register is set then SYS_RESET# will result in a full power cycle reset.

#### 5.14.9.5 THRMTRIP# Signal

If THRMTRIP# goes active, the processor is indicating an overheat condition, and Chipset immediately transitions to an S5 state. However, since the processor has overheated, it does not respond to chipset’s STPCLK# pin with a stop grant special cycle. Therefore, Chipset does not wait for one. Immediately upon seeing THRMTRIP# low, Chipset initiates a transition to the S5 state, drive SLP_S3#, SLP_S4#, SLP_S5# low, and set the CTS bit. The transition looks like a power button override.

It is extremely important that when a THRMTRIP# event occurs, Chipset power down immediately without following the normal S0 -> S5 path. This path may be taken in parallel, but Chipset must immediately enter a power down state. It does this by driving SLP_S3#, SLP_S4#, and SLP_S5# immediately after sampling THRMTRIP# active.

If the processor is running extremely hot and is heating up, it is possible (although very unlikely) that components around it, such as Chipset, are no longer executing cycles properly. Therefore, if THRMTRIP# goes active, and Chipset is relying on state machine logic to perform the power down, the state machine may not be working, and the system will not power down.

Chipset follows this flow for THRMTRIP#.

1. At boot (PLTRST# low), THRMTRIP# ignored.
2. After power-up (PLTRST# high), if THRMTRIP# sampled active, SLP_S3#, SLP_S4#, and SLP_S5# assert, and normal sequence of sleep machine starts.
3. Until sleep machine enters the S5 state, SLP_S3#, SLP_S4#, and SLP_S5# stay active, even if THRMTRIP# is now inactive. This is the equivalent of “latching” the thermal trip event.
4. If S5 state reached, go to step #1, otherwise stay here. If Chipset does not reach S5, Chipset does not reboot until power is cycled.

During boot, THRMTRIP# is ignored until SLP_S3#, PWROK, VRMPWRGD/VGATE, and PLTRST# are all '1'. During entry into a powered-down state (due to S3, S4, S5 entry, power cycle reset, etc.) THRMTRIP# is ignored until either SLP_S3# = 0, or PWROK = 0, or VRMPWRGD/VGATE = 0.

***Note:*** A thermal trip event will:

- Set the AFTERG3_EN bit
- Clear the PWRBTN_STS bit
- Clear all the GPE0_EN register bits
- Clear the SMB_WAK_STS bit only if SMB_SAK_STS was set due to SMBus slave receiving message and not set due to SMBAlert

#### 5.14.9.6 BM_BUSY# (Netbook Only)

The BM_BUSY# signal is an input from a graphics component to indicate if it is busy. If prior to going to the C3 state, the BM_BUSY# signal is active, then the BM_STS bit will be set. If after going to the C3 state, the BM_BUSY# signal goes back active, Chipset will treat this as if one of the PCI REQ# signals went active. This is treated as a break event.

### 5.14.10 ALT Access Mode

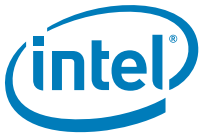
Before entering a low power state, several registers from powered down parts may need to be saved. In the majority of cases, this is not an issue, as registers have read and write paths. However, several of the ISA compatible registers are either read only or write only. To get data out of
write-only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

If the ALT access mode is entered and exited after reading the registers of Chipset 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 Chipset 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. For example DOS and its associated software assume that the system timer is running at 54.6 ms and as a result the time-outs in the software may be happening faster than expected.

Operating systems (e.g., Microsoft Windows* 98, Windows* 2000, and Windows NT*) reprogram the system timer and therefore do not encounter this problem.



For some other loss (e.g., Microsoft MS-DOS*) 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.

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

The registers described in Table 5-64 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.

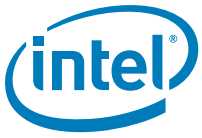
**Table 5-64.Write Only Registers with Read Paths in ALT Access Mode (Sheet 1 of 2)**

| Restore Data | | | | Restore Data | | | |
|---|---|---|---|---|---|---|---|
| I/O Addr | # of Rds | Access | Data | I/O Addr | # of Rds | Access | Data |
| 00h | 2 | 1 | DMA Chan 0 base address low byte | 40h | 7 | 1 | Timer Counter 0 status, bits [5:0] |
| | | 2 | DMA Chan 0 base address high byte | | | 2 | Timer Counter 0 base count low byte |
| 01h | 2 | 1 | DMA Chan 0 base count low byte | | | 3 | Timer Counter 0 base count high byte |
| | | 2 | DMA Chan 0 base count high byte | | | 4 | Timer Counter 1 base count low byte |
| 02h | 2 | 1 | DMA Chan 1 base address low byte | | | 5 | Timer Counter 1 base count high byte |
| | | 2 | DMA Chan 1 base address high byte | | | 6 | Timer Counter 2 base count low byte |
| 03h | 2 | 1 | DMA Chan 1 base count low byte | | | 7 | Timer Counter 2 base count high byte |
| | | 2 | DMA Chan 1 base count high byte | 41h | 1 | | Timer Counter 1 status, bits [5:0] |
| 04h | 2 | 1 | DMA Chan 2 base address low byte | 42h | 1 | | Timer Counter 2 status, bits [5:0] |
| | | 2 | DMA Chan 2 base address high byte | 70h | 1 | | Bit 7 = NMI Enable, Bits [6:0] = RTC Address |
| 05h | 2 | 1 | DMA Chan 2 base count low byte | C4h | 2 | 1 | DMA Chan 5 base address low byte |
| | | 2 | DMA Chan 2 base count high byte | | | 2 | DMA Chan 5 base address high byte |
| 06h | 2 | 1 | DMA Chan 3 base address low byte | C6h | 2 | 1 | DMA Chan 5 base count low byte |
| | | 2 | DMA Chan 3 base address high byte | | | 2 | DMA Chan 5 base count high byte |

### Table 5-64.Write Only Registers with Read Paths in ALT Access Mode (Sheet 2 of 2)

| Restore Data | | | | Restore Data | | | |
|---|---|---|---|---|---|---|---|
| I/O Addr | # of Rds | Access | Data | I/O Addr | # of Rds | Access | Data |
| 07h | 2 | 1 | DMA Chan 3 base count low byte | C8h | 2 | 1 | DMA Chan 6 base address low byte |
| | | 2 | DMA Chan 3 base count high byte | | | 2 | DMA Chan 6 base address high byte |
| 08h | 6 | 1 | DMA Chan 0–3 Command[2] | CAh | 2 | 1 | DMA Chan 6 base count low byte |
| | | 2 | DMA Chan 0–3 Request | | | 2 | DMA Chan 6 base count high byte |
| | | 3 | DMA Chan 0 Mode: Bits(1:0) = 00 | CCh | 2 | 1 | DMA Chan 7 base address low byte |
| | | 4 | DMA Chan 1 Mode: Bits(1:0) = 01 | | | 2 | DMA Chan 7 base address high byte |
| | | 5 | DMA Chan 2 Mode: Bits(1:0) = 10 | CEh | 2 | 1 | DMA Chan 7 base count low byte |
| | | 6 | DMA Chan 3 Mode: Bits(1:0) = 11. | | | 2 | DMA Chan 7 base count high byte |
| 20h | 12 | 1 | PIC ICW2 of Master controller | D0h | 6 | 1 | DMA Chan 4–7 Command[1] |
| | | 2 | PIC ICW3 of Master controller | | | 2 | DMA Chan 4–7 Request |
| | | 3 | PIC ICW4 of Master controller | | | 3 | DMA Chan 4 Mode: Bits(1:0) = 00 |
| | | 4 | PIC OCW1 of Master controller[2] | | | 4 | DMA Chan 5 Mode: Bits(1:0) = 01 |
| | | 5 | PIC OCW2 of Master controller | | | 5 | DMA Chan 6 Mode: Bits(1:0) = 10 |
| | | 6 | PIC OCW3 of Master controller | | | 6 | DMA Chan 7 Mode: Bits(1:0) = 11. |
| | | 7 | PIC ICW2 of Slave controller | | | | |
| | | 8 | PIC ICW3 of Slave controller | | | | |
| | | 9 | PIC ICW4 of Slave controller | | | | |
| | | 10 | PIC OCW1 of Slave controller[1] | | | | |
| | | 11 | PIC OCW2 of Slave controller | | | | |
| | | 12 | PIC OCW3 of Slave controller | | | | |

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



#### 5.14.10.2 PIC Reserved Bits

Many bits within the PIC are reserved, and must have certain values written in order for the PIC to operate properly. Therefore, there is no need to return these values in ALT access mode. When reading PIC registers from 20h and A0h, the reserved bits shall return the values listed in Table 5-65.

**Table 5-65. PIC Reserved Bits Return Values**

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

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

The registers described in Table 5-66 have write paths to them in ALT access mode. Software restores these values after returning from a powered down state. These registers must be handled special by software. When in normal mode, writing to the base address/count register also writes to the current address/count register. Therefore, the base address/count must be written first, then the part is put into ALT access mode and the current address/count register is written.

**Table 5-66. Register Write Accesses in ALT Access Mode**

| I/O Address | Register Write Value |
|---|---|
| 08h | DMA Status Register for channels 0–3. |
| D0h | DMA Status Register for channels 4–7. |

### 5.14.11 System Power Supplies, Planes, and Signals

#### 5.14.11.1 Power Plane Control with SLP_S3#, SLP_S4# and SLP_S5#

The usage of SLP_S3# and SLP_S4# depends on whether the platform is configured for $S3_{HOT}$ and $S3_{COLD}$.

##### $S3_{HOT}$

The SLP_S3# output signal is used to cut power only to the processor and associated subsystems and to optionally stop system clocks.

### $S3_{COLD}$

The SLP_S3# output signal can be used to cut power to the system core supply, since it only goes active for the STR state (typically mapped to ACPI S3). Power must be maintained to Chipset resume well, and to any other circuits that need to generate Wake signals from the STR state.

Cutting power to the core may be done via the power supply, or by external FETs to the motherboard.

The SLP_S4# or SLP_S5# output signal can be used to cut power to the system core supply, as well as power to the system memory, since the context of the system is saved on the disk. Cutting power to the memory may be done via the power supply, or by external FETs to the motherboard.

The SLP_S4# output signal is used to remove power to additional subsystems that are powered during SLP_S3#.

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

#### 5.14.11.2 SLP_S4# and Suspend-To-RAM Sequencing

The system memory suspend voltage regulator is controlled by the Glue logic. The SLP_S4# signal should be used to remove power to system memory rather than the SLP_S5# signal. The SLP_S4# logic in Chipset provides a mechanism to fully cycle the power to the DRAM and/or detect if the power is not cycled for a minimum time.

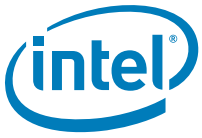
***Note:*** To utilize the minimum DRAM power-down feature that is enabled by the SLP_S4# Assertion Stretch Enable bit (D31:F0:A4h bit 3), the DRAM power must be controlled by the SLP_S4# signal.

#### 5.14.11.3 PWROK Signal

The PWROK input should go active based on the core supply voltages becoming valid. PWROK should go active no sooner than 100 ms after Vcc3_3 and Vcc1_5 have reached their nominal values.

***Note:***

1. SYS_RESET# is recommended for implementing the system reset button. This saves external logic that is needed if the PWROK input is used. Additionally, it allows for better handling of the SMBus and processor resets, and avoids improperly reporting power failures.
2. If the PWROK input is used to implement the system reset button, Chipset does not provide any mechanism to limit the amount of time that the processor is held in reset. The platform must externally ensure that maximum reset assertion specifications are met.



3. If a design has an active-low reset button electrically AND'd with the PWROK signal from the power supply and the processor's voltage regulator module Chipset PWROK_FLR bit will be set. Chipset treats this internally as if the RSMRST# signal had gone active. However, it is not treated as a full power failure. If PWROK goes inactive and then active (but RSMRST# stays high), then Chipset reboots (regardless of the state of the AFTERG3 bit). If the RSMRST# signal also goes low before PWROK goes high, then this is a full power failure, and the reboot policy is controlled by the AFTERG3 bit.
4. PWROK and RSMRST# are sampled using the RTC clock. Therefore, low times that are less than one RTC clock period may not be detected by Chipset.
5. In the case of true PWROK failure, PWROK goes low first before the VRMPWRGD.

#### 5.14.11.4 CPUPWRGD Signal

This signal is connected to the processor's VRM via the VRMPWRGD signal and is internally AND'd with the PWROK signal that comes from the system power supply.

#### 5.14.11.5 VRMPWRGD Signal

VRMPWRGD is an input from the regulator indicating that all of the outputs from the regulator are on and within specification. VRMPWRGD may go active before or after the PWROK from the main power supply. Chipset has no dependency on the order in which these two signals go active or inactive.

#### 5.14.11.6 BATLOW# (Battery Low) (Netbook Only)

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

#### 5.14.11.7 Controlling Leakage and Power Consumption during Low-Power States

To control leakage in the system, various signals tri-state or go low during some low-power states.

General principles:

- All signals going to powered down planes (either internally or externally) must be either tri-stated or driven low.
- Signals with pull-up resistors should not be low during low-power states. This is to avoid the power consumed in the pull-up resistor.
- Buses should be halted (and held) in a known state to avoid a floating input (perhaps to some other device). Floating inputs can cause extra power consumption.

Based on the above principles, the following measures are taken:

- During S3 (STR), all signals attached to powered down planes are tri-stated or driven low.

## 5.14.12 Clock Generators

The clock generator is expected to provide the frequencies shown in Table 5-67.

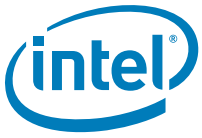
**Table 5-67.Chipset Clock Inputs**

| Clock Domain | Frequency | Source | Usage |
|---|---|---|---|
| SATA_CLK | 100 MHz Differential | Main Clock Generator | Used by SATA controller. Stopped in S3 ~ S5 based on SLP_S3# assertion. |
| DMI_CLK | 100 MHz Differential | Main Clock Generator | Used by DMI and PCI Express*. Stopped in S3 ~ S5 based on SLP_S3# assertion. |
| PCICLK | 33 MHz | Main Clock Generator | Nettop Only: Free-running PCI Clock to Chipset. Stopped in S3 ~ S5 based on SLP_S3# assertion.<br>Netbook Only: Free-running (not affected by STP_PCI# PCI Clock to Chipset. This is not the system PCI clock. This clock must keep running in S0 while the system PCI clock may stop based on CLKRUN# protocol. Stopped in S3 ~ S5 based on SLP_S3# assertion. |
| CLK48 | 48.000 MHz | Main Clock Generator | Used by USB controllers and Intel HD Audio controller. Stopped in S3 ~ S5 based on SLP_S3# assertion. |
| CLK14 | 14.318 MHz | Main Clock Generator | Used by ACPI timers. Stopped in S3 ~ S5 based on SLP_S3# assertion. |
| LAN_CLK | 0.8 to 50 MHz | LAN Connect | LAN Connect Interface. Control policy is determined by the clock source. |

### 5.14.12.1 Clock Control Signals from Chipset to Clock Synthesizer (Netbook Only)

The clock generator is assumed to have a direct connection from the following Chipset signals:

- STP_CPU#: Stops processor clocks in C3 and C4 states
- STP_PCI#: Stops system PCI clocks (not Chipset free-running 33 MHz clock) due to CLKRUN# protocol
- SLP_S3#: Expected to drive clock chip PWRDOWN (through inverter), to stop clocks in $S3_{HOT}$ and on the way to $S3_{COLD}$ to S5.



## 5.14.13 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. Chipset does not support burst modes.

### 5.14.13.1 APM Power Management (Nettop Only)

Chipset has a timer that, when enabled by the 1MIN_EN bit in the SMI Control and Enable register, generates an SMI# once per minute. The SMI handler can check for system activity by reading the DEVACT_STS register. If none of the system bits are set, the SMI handler can increment a software counter. When the counter reaches a sufficient number of consecutive minutes with no activity, the SMI handler can then put the system into a lower power state.

If there is activity, various bits in the DEVACT_STS register will be set. Software clears the bits by writing a 1 to the bit position.

The DEVACT_STS register allows for monitoring various internal devices, or Super I/O devices (SP, PP, FDC) on LPC or PCI, keyboard controller accesses, or audio functions on LPC or PCI. Other PCI activity can be monitored by checking the PCI interrupts.

### 5.14.13.2 Mobile APM Power Management (Netbook Only)

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

1. The periodic SMI# timer checks if a device is idle for the required 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

The SMI# handler exits with an I/O restart. This allows the original software to continue.

## 5.15 System Management (D31:F0)

Chipset provides various functions to make a system easier to manage and to lower the Total Cost of Ownership (TCO) of the system. In addition, Chipset provides integrated ASF Management support. Features and functions can be augmented via external A/D converters and GPIO, as well as an external microcontroller.

The following features and functions are supported by Chipset:

- Processor present detection
  - Detects if processor fails to fetch the first instruction after reset
- Various Error detection (such as ECC Errors) Indicated by host controller
  - Can generate SMI#, SCI, SERR, NMI, or TCO interrupt
- Intruder Detect input
  - Can generate TCO interrupt or SMI# when the system cover is removed
  - INTRUDER# allowed to go active in any power state, including G3
- Detection of bad Firmware Hub programming
  - Detects if data on first read is FFh (indicates unprogrammed Firmware Hub)
- Ability to hide a PCI device
  - Allows software to hide a PCI device in terms of configuration space through the use of a device hide register (See Section 10.1.52)
- Integrated ASF Management support

***Note:*** Voltage ID from the processor can be read via GPI signals.

### 5.15.1 Theory of Operation

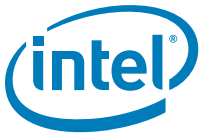
The System Management functions are designed to allow the system to diagnose failing subsystems. The intent of this logic is that some of the system management functionality be provided without the aid of an external microcontroller.

#### 5.15.1.1 Detecting a System Lockup

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

#### 5.15.1.2 Handling an Intruder

Chipset has an input signal, INTRUDER#, that can be attached to a switch that is activated by the system's case being open. This input has a two RTC clock debounce. If INTRUDER# goes active (after the debouncer), this will set the INTRD_DET bit in the TCO_STS register. The INTRD_SEL bits in the TCO_CNT register can enable Chipset 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 signal will go to a 0 when INTRUDER# input signal goes inactive. Note that this is slightly different than a classic sticky bit, since most sticky bits would remain active indefinitely when the signal goes active and would immediately go inactive when a 1 is written to the bit.

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

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

#### 5.15.1.3 Detecting Improper Firmware Hub Programming

Chipset can detect the case where the Firmware Hub is not programmed. This results in the first instruction fetched to have a value of FFh. If this occurs, Chipset sets the BAD_BIOS bit, which can then be reported via the Heartbeat and Event reporting using an external, Alert on LAN* enabled LAN controller (See Section 5.15.2).

### 5.15.2 Heartbeat and Event Reporting via SMBus

Chipset integrated LAN controller supports ASF heartbeat and event reporting functionality when used with the 82562EM or 82562EX Platform LAN Connect component. This allows the integrated LAN controller to report messages to a network management console without the aid of the system processor. This is crucial in cases where the processor is malfunctioning or cannot function due to being in a low-power state.

All heartbeat and event messages are sent on the SMBus interface. This allows an external LAN controller to act upon these messages if the internal LAN controller is not used.

The basic scheme is for Chipset integrated LAN controller to send a prepared Ethernet message to a network management console. The prepared message is stored in the non-volatile EEPROM that is connected to Chipset.

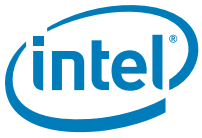
Messages are sent by the LAN controller either because a specific event has occurred, or they are sent periodically (also known as a heartbeat). The event and heartbeat messages have the exact same format. The event messages are sent based on events

occurring. The heartbeat messages are sent every 30 to 32 seconds. When an event occurs, Chipset sends a new message and increments the SEQ[3:0] field. For heartbeat messages, the sequence number does not increment.

The following rules/steps apply if the system is in a G0 state and the policy is for Chipset **to reboot** the system after a hardware lockup:

1. On detecting the lockup, the SECOND_TO_STS bit is set. Chipset may send up to 1 Event message to the LAN controller. Chipset then attempts to reboot the processor.
2. If the reboot at step 1 is successful then the BIOS should clear the SECOND_TO_STS bit. This prevents any further Heartbeats from being sent. The BIOS may then perform addition recovery/boot steps. (See note 2, below.)
3. If the reboot attempt in step 1 is not successful, the timer will timeout a third time. At this point the system has locked up and was unsuccessful in rebooting. Chipset does not attempt to automatically reboot again. Chipset starts sending a message every heartbeat period
(30–32 seconds). The heartbeats continue until some external intervention occurs (reset, power failure, etc.).
4. After step 3 (unsuccessful reboot after third timeout), if the user does a Power Button Override, the system goes to an S5 state. Chipset continues sending the messages every heartbeat period.
5. After step 4 (power button override after unsuccessful reboot) if the user presses the Power Button again, the system should wake to an S0 state and the processor should start executing the BIOS.
6. If step 5 (power button press) is successful in waking the system, Chipset continues sending messages every heartbeat period until the BIOS clears the SECOND_TO_STS bit. (See note 2)
7. If step 5 (power button press) is unsuccessful in waking the system, Chipset continues sending a message every heartbeat period. Chipset does not attempt to automatically reboot again. Chipset starts sending a message every heartbeat period (30–32 seconds). The heartbeats continue until some external intervention occurs (reset, power failure, etc.).
(See note 3)
8. After step 3 (unsuccessful reboot after third timeout), if a reset is attempted (using a button that pulses PWROK low or via the message on the SMBus slave I/F), Chipset attempts to reset the system.
9. After step 8 (reset attempt) if the reset is successful, the BIOS is run. Chipset continues sending a message every heartbeat period until the BIOS clears the SECOND_TO_STS bit. (See note 2)
10. After step 8 (reset attempt), if the reset is unsuccessful, Chipset continues sending a message every heartbeat period. Chipset does not attempt to reboot the system again without external intervention. (See note 3)

The following rules/steps apply if the system is in a G0 state and the policy is for Chipset to **not reboot** the system after a hardware lockup.



1. On detecting the lockup the SECOND_TO_STS bit is set. Chipset sends a message with the Watchdog (WD) Event status bit set (and any other bits that must also be set). This message is sent as soon as the lockup is detected, and is sent with the next (incremented) sequence number.
2. After step 1, Chipset sends a message every heartbeat period until some external intervention occurs.
3. Rules/steps 4–10 apply if no user intervention (resets, power button presses, SMBus reset messages) occur after a third timeout of the watchdog timer. If the intervention occurs before the third timeout, then jump to rule/step 11.
4. After step 3 (third timeout), if the user does a Power Button Override, the system goes to an S5 state. Chipset continues sending heartbeats at this point.
5. After step 4 (power button override), if the user presses the power button again, the system should wake to an S0 state and the processor should start executing the BIOS.
6. If step 5 (power button press) is successful in waking the system, Chipset continues sending heartbeats until the BIOS clears the SECOND_TO_STS bit. (See note 2)
7. If step 5 (power button press) is unsuccessful in waking the system, Chipset continues sending heartbeats. Chipset does not attempt to reboot the system again until some external intervention occurs (reset, power failure, etc.). (See note 3)
8. After step 3 (third timeout), if a reset is attempted (using a button that pulses PWROK low or via the message on the SMBus slave I/F), Chipset attempts to reset the system.
9. If step 8 (reset attempt) is successful, the BIOS is run. Chipset continues sending heartbeats until the BIOS clears the SECOND_TO_STS bit. (See note 2)
10. If step 8 (reset attempt), is unsuccessful, Chipset continues sending heartbeats. Chipset does not attempt to reboot the system again without external intervention. **Note:** A system that has locked up and can not be restarted with power button press is probably broken (bad power supply, short circuit on some bus, etc.)
11. This and the following rules/steps apply if the user intervention (power button press, reset, SMBus message, etc.) occur prior to the third timeout of the watchdog timer.
12. After step 1 (second timeout), if the user does a Power Button Override, the system goes to an S5 state. Chipset continues sending heartbeats at this point.
13. After step 12 (power button override), if the user presses the power button again, the system should wake to an S0 state and the processor should start executing the BIOS.
14. If step 13 (power button press) is successful in waking the system, Chipset continues sending heartbeats until the BIOS clears the SECOND_TO_STS bit. (See note 2)
15. If step 13 (power button press) is unsuccessful in waking the system, Chipset continues sending heartbeats. Chipset does not attempt to reboot the system again until some external intervention occurs (reset, power failure, etc.). (See note 3)

16. After step 1 (second timeout), if a reset is attempted (using a button that pulses PWROK low or via the message on the SMBus slave I/F), Chipset attempts to reset the system.
17. If step 16 (reset attempt) is successful, the BIOS is run. Chipset continues sending heartbeats until the BIOS clears the SECOND_TO_STS bit. (See note 2)
18. If step 16 (reset attempt), is unsuccessful, Chipset continues sending heartbeats. Chipset does not attempt to reboot the system again without external intervention. (See note 3)

If the system is in a G1 (S1–S4) state, Chipset sends a heartbeat message every 30–32 seconds. If an event occurs prior to the system being shutdown, Chipset immediately sends an event message with the next incremented sequence number. After the event message, Chipset resumes sending heartbeat messages.

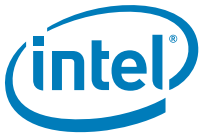
***Note:*** Notes for previous two numbered lists.

1. Normally, Chipset does not send heartbeat messages while in the G0 state (except in the case of a lockup). However, if a hardware event (or heartbeat) occurs just as the system is transitioning into a G0 state, the hardware continues to send the message even though the system is in a G0 state (and the status bits may indicate this).

   These messages are sent via the SMBus. Chipset abides by the SMBus rules associated with collision detection. It delays starting a message until the bus is idle, and detects collisions. If a collision is detected Chipset waits until the bus is idle, and tries again.
2. **WARNING:** It is important the BIOS clears the SECOND_TO_STS bit, as the alerts interfere with the LAN device driver from working properly. The alerts reset part of the LAN controller and would prevent an operating system's device driver from sending or receiving some messages.
3. A system that has locked up and can not be restarted with power button press is assumed to have broken hardware (bad power supply, short circuit on some bus, etc.), and is beyond chipset's recovery mechanisms.
4. A spurious alert could occur in the following sequence:
   — The processor has initiated an alert using the SEND_NOW bit
   — During the alert, the THRM#, INTRUDER# or GPIO11 changes state
   — The system then goes to a non-S0 state.

Once the system transitions to the non-S0 state, it may send a single alert with an incremental SEQUENCE number.

5. An inaccurate alert message can be generated in the following scenario
   — The system successfully boots after a second watchdog Timeout occurs.
   — PWROK goes low (typically due to a reset button press) or a power button override occurs (before the SECOND_TO_STS bit is cleared).



— An alert message indicating that the processor is missing or locked up is generated with a new sequence number.

Table 5-68 shows the data included in the Alert on LAN messages.

**Table 5-68.Heartbeat Message Data**

| Field | Comment |
|---|---|
| Cover Tamper Status | 1 = This bit is set if the intruder detect bit is set (INTRD_DET). |
| Temp Event Status | 1 = This bit is set if Chipset THERM# input signal is asserted. |
| Processor Missing Event Status | 1 = This bit is set if the processor failed to fetch its first instruction. |
| TCO Timer Event Status | 1 = This bit is set when the TCO timer expires. |
| Software Event Status | 1 = This bit is set when software writes a 1 to the SEND_NOW bit. |
| Unprogrammed Firmware Hub Event Status | 1 = First BIOS fetch returned a value of FFh, indicating that the Firmware Hub has not yet been programmed (still erased). |
| GPIO Status | 1 = This bit is set when GPIO11 signal is high.<br>0 = This bit is cleared when GPIO11 signal is low.<br>An event message is triggered on an transition of GPIO11. |
| SEQ[3:0] | This is a sequence number. It initially is 0, and increments each time Chipset sends a new message. Upon reaching 1111, the sequence number rolls over to 0000. MSB (SEQ3) sent first. |
| System Power State | 00 = G0, 01 = G1, 10 = G2, 11 = Pre-Boot. MSB sent first |
| MESSAGE1 | Will be the same as the MESSAGE1 Register. MSB sent first. |
| MESSAGE2 | Will be the same as the MESSAGE2 Register. MSB sent first. |
| WDSTATUS | Will be the same as the WDSTATUS Register. MSB sent first. |

## 5.16 SATA Host Controller (D31:F2)

The SATA function in Chipset has dual modes of operation to support different operating system conditions. In the case of Native IDE enabled operating systems, Chipset unitize single controllers to enable two ports of the bus.

The MAP register, Section 15.1.33, provides the ability to share PCI functions. When sharing is enabled, all decode of I/O is done through the SATA registers. Device 31, Function 1 (IDE controller) is hidden by software writing to the Function Disable Register (D31, F0, offset F2h, bit 1), and its configuration registers are not used.

Chipset SATA controller features two sets of interface signals (ports) that can be independently enabled or disabled (they cannot be tri-stated or driven low). Each interface is supported by an independent DMA controller.

Chipset SATA controller interacts with an attached mass storage device through a register interface that is equivalent to that presented by a traditional IDE host adapter. The host software follows existing standards and conventions when accessing the register interface and follows standard command protocol conventions.

Table 5-69 lists Chipset SATA Feature support information. Table 5-70 contains descriptions for the SATA features listed in Table 5-69.

***Note:*** SATA interface transfer rates are independent of UDMA mode settings. SATA interface transfer rates will operate at the bus's maximum speed, regardless of the UDMA mode reported by the SATA device or the system BIOS.

**Table 5-69.SATA Features Support in Chipset**

| Feature | Chipset (AHCI Disabled) | Chipset (AHCI Enabled) |
|---|---|---|
| Native Command Queuing (NCQ) | N/A | Supported |
| Auto Activate for DMA | N/A | Supported |
| Hot Plug Support | N/A | Supported |
| Asynchronous Signal Recovery | N/A | Supported |
| 3 Gb/s Transfer Rate | Supported | Supported |
| ATAPI Asynchronous Notification | N/A | Supported |
| Host Initiated Power Management | N/A | Supported (Netbook Only) |
| Staggered Spin-Up | Supported | Supported |
| Command Completion Coalescing | N/A | N/A |
| Port Multiplier | N/A | N/A |
| External SATA | N/A | N/A |

**Table 5-70.SATA Feature Description**

| Feature | Description |
|---|---|
| Native Command Queuing (NCQ) | Allows the device to reorder commands for more efficient data transfers |
| Auto Activate for DMA | Collapses a DMA Setup then DMA Activate sequence into a DMA Setup only |
| Hot Plug Support | Allows for device detection without power being applied and ability to connect and disconnect devices without prior notification to the system |
| Asynchronous Signal Recovery | Provides a recovery from a loss of signal or establishing communication after hot plug |
| 3 Gb/s Transfer Rate | Capable of data transfers up to 3Gb/s |
| ATAPI Asynchronous Notification | A mechanism for a device to send a notification to the host that the device requires attention |
| Host Initiated Power Management | Capability for the host controller to request Partial and Slumber interface power states |

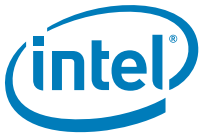


**Table 5-70.SATA Feature Description**

| | |
|---|---|
| Staggered Spin-Up | Enables the host the ability to spin up hard drives sequentially to prevent power load problems on boot |
| Command Completion Coalescing | Reduces interrupt and completion overhead by allowing a specified number of commands to complete and then generating an interrupt to process the commands |
| Port Multiplier | A mechanism for one active host connection to communicate with multiple devices |
| External SATA | Technology that allows for an outside the box connection of up to 2 meters (when using the cable defined in SATA-IO) |

## 5.16.1 Theory of Operation

### 5.16.1.1 Standard ATA Emulation

Chipset contains a set of registers that shadow the contents of the legacy IDE registers. The behavior of the Command and Control Block registers, PIO, and DMA data transfers, resets, and interrupts are all emulated.

***Note:*** Chipset will assert INTR when the master device completes the EDD command regardless of the command completion status of the slave device. If the master completes EDD first, an INTR is generated and BSY will remain '1' until the slave completes the command. If the slave completes EDD first, BSY will be '0' when the master completes the EDD command and asserts INTR. Software must wait for busy to clear (0) before completing an EDD command, as required by the ATA5 through ATA7 (T13) industry standards.

### 5.16.1.2 48-Bit LBA Operation

The SATA host controller supports 48-bit LBA through the host-to-device register FIS when accesses are performed via writes to the task file. The SATA host controller will ensure that the correct data is put into the correct byte of the host-to-device FIS.

There are special considerations when reading from the task file to support 48-bit LBA operation. Software may need to read all 16-bits. Since the registers are only 8-bits wide and act as a FIFO, a bit must be set in the device/control register, which is at offset 3F6h for primary and 376h for secondary (or their native counterparts).

If software clears bit 7 of the control register before performing a read, the last item written will be returned from the FIFO. If software sets bit 7 of the control register before performing a read, the first item written will be returned from the FIFO.

### 5.16.2 SATA Swap Bay Support

Dynamic Hot-Plug (e.g., surprise removal) is not supported by the SATA host controller without special support from AHCI and the proper board hardware. However, Chipset does provide for basic SATA swap bay support using the PSC register configuration bits and power management flows. A device can be powered down by software and the port can then be disabled, allowing removal and insertion of a new device.

***Note:*** This SATA swap bay operation requires board hardware (implementation specific), BIOS, and operating system support.

### 5.16.3 Power Management Operation

Power management of Chipset SATA controller and ports will cover operations of the host controller and the SATA wire.

#### 5.16.3.1 Power State Mappings

The D0 PCI power management state for device is supported by Chipset SATA controller.

SATA devices may also have multiple power states. From parallel ATA, 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 via the STANDBY IMMEDIATE command. However, an ACPI method is also called which will reset the device and then cut its power.

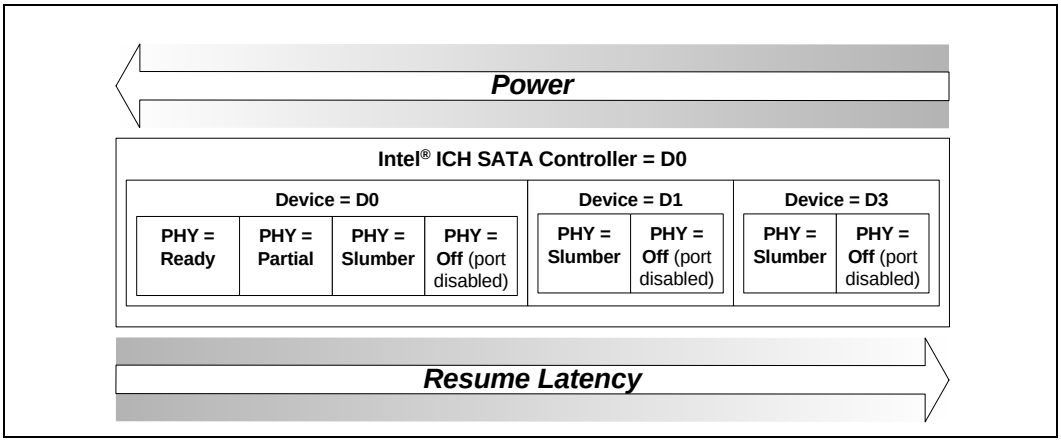
Each of these device states are subsets of the host controller's D0 state.

Finally, SATA 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 active
- **Partial** – PHY logic is powered, but in a reduced state. Exit latency is no longer than 10 ns
- **Slumber** – PHY logic is powered, but in a reduced state. Exit latency can be up to 10 ms.

Since these states have much lower exit latency than the ACPI D1 and D3 states, the SATA controller defines these states as sub-states of the device D0 state.

Figure 5-13. SATA Power States




### 5.16.3.2 Power State Transitions

#### 5.16.3.2.1 Partial and Slumber State Entry/Exit

The partial and slumber states save interface power when the interface is idle. It would be most analogous to PCI 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 via 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 COM_WAKE to bring the link back online. Similarly, the SATA device must perform the same action.

#### 5.16.3.2.2 Device D1, 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 then sending commands over the interface to the device. The command most likely to be used in ATA/ATAPI is the “STANDBY IMMEDIATE” command.

#### 5.16.3.2.3 Host Controller $D3_{HOT}$ State

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

1. When the power state is D3, only accesses to configuration space are allowed. Any attempt to access the memory or I/O spaces will result in master abort.

2. When the power state is D3, no interrupts may be generated, even if they are enabled. If an interrupt status bit is pending when the controller transitions to D0, an interrupt may be generated.

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

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

#### 5.16.3.2.4 Non-AHCI Mode PME# Generation

When in non-AHCI mode (legacy mode) of operation, the SATA controller does not generate PME#. This includes attach events (since the port must be disabled), or interlock switch events (via the SATAGP pins).

### 5.16.3.3 SMI Trapping (APM)

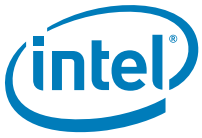
Device 31:Function2:Offset C0h (see Section 15.1.40) contain control for generating SMI# on accesses to the IDE I/O spaces. These bits map to the legacy ranges (1F0–1F7h, 3F6h, 170–177h, and 376h). If the SATA controller is in legacy mode and is using these addresses, accesses to one of these ranges with the appropriate bit set causes the cycle to not be forwarded to the SATA controller, and for an SMI# to be generated. If an access to the Bus-Master IDE registers occurs while trapping is enabled for the device being accessed, then the register is updated, an SMI# is generated, and the device activity status bits (Section 15.1.41) are updated indicating that a trap occurred.

## 5.16.4 SATA LED

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

## 5.16.5 AHCI Operation

Chipset provides hardware support for Advanced Host Controller Interface (AHCI), a programming interface for SATA host controllers developed thru a joint industry effort. AHCI defines transactions between the SATA controller and software and enables advanced performance and usability with SATA. Platforms supporting AHCI may take advantage of performance features such as no master/slave designation for SATA devices—each device is treated as a master—and hardware assisted native command queuing. AHCI also provides usability enhancements such as Hot-Plug. AHCI requires appropriate software support (e.g., an AHCI driver) and for some features, hardware support in the SATA device or additional platform hardware.



Chipset supports all of the mandatory features of the *Serial ATA Advanced Host Controller Interface Specification*, Revision 1.0 and many optional features, such as hardware assisted native command queuing, aggressive power management, LED indicator support, and Hot-Plug thru the use of interlock switch support (additional platform hardware and software may be required depending upon the implementation).

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

***Note:*** When there are more than two PRD entries for a PIO data transfer that spans multiple DATA FISes, Chipset does not support intermediate PRD entries that are less than 144 Words in size when Chipset is operating in AHCI mode at 1.5 Gb/s.

## 5.17 High Precision Event Timers

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

Chipset provides three timers. The three timers are implemented as a single counter each with its own comparator and value register. This counter increases monotonically. Each individual timer can generate an interrupt when the value in its value register matches the value in the main counter.

The registers associated with these timers are mapped to a 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. The hardware can support an assignable decode space; however, the BIOS sets this space prior to handing it over to the operating system
(See Section 9.4). It is not expected that the operating system will move the location of these timers once it is set by the BIOS.

### 5.17.1 Timer Accuracy

1. The timers are accurate over any 1 ms period to within 0.05% of the time specified in the timer resolution fields.
2. Within any 100 microsecond period, the timer reports a time that is up to two ticks too early or too late. Each tick is less than or equal to 100 ns, so this represents an error of less than 0.2%.
3. 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 is clocked by the 14.31818 MHz clock, synchronized into the 66.666 MHz domain. This results in a non-uniform duty cycle on the synchronized clock, but does have the correct average period. The accuracy of the main counter is as accurate as the 14.3818 MHz clock.

## 5.17.2 Interrupt Mapping

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

In this case, the Legacy Replacement Rout bit (LEG_RT_CNF) is set. This forces the mapping found in Table 5-71.

**Table 5-71.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 | Per IRQ Routing Field. | Per IRQ Routing Field | |

### 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 supported interrupt values are IRQ 20, 21, 22, and 23.

## 5.17.3 Periodic vs. Non-Periodic Modes

### Non-Periodic Mode

Timer 0 is configurable to 32 (default) or 64-bit mode, whereas Timers 1 and 2 only support 32-bit mode (See Section 20.1.5).

All three timers support non-periodic mode.

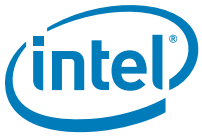
Consult Section 2.3.9.2.1 of the IA-PC HPET Specification for a description of this mode.

### Periodic Mode

Timer 0 is the only timer that supports periodic mode. Consult Section 2.3.9.2.2 of the *IA-PC HPET Specification* for a description of this mode.

The following usage model is expected:

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



4. Software writes the new value in the TIMER0_COMPARATOR_VAL register
5. Software sets the ENABLE_CNF bit to enable interrupts.

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

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

### 5.17.4 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), 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 04h, bit 0).
2. Set the timer type field (selects one-shot or periodic).
3. Set the interrupt enable
4. Set the comparator value

### 5.17.5 Interrupt Levels

Interrupts directed to the internal 8259s are active high. See Section 5.10 for information regarding the polarity programming of the I/O APIC for detecting internal interrupts.

If the interrupts are mapped to the I/O APIC and set for level-triggered mode, they can be shared with PCI interrupts. This may be shared although it's 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.

### 5.17.6 Handling Interrupts

If each timer has a unique interrupt and the timer has been configured for edge-triggered mode, then there are no specific steps required. No read is required to process the interrupt.

If a timer has been configured to level-triggered mode, then its interrupt must be cleared by the software. This is done by reading the interrupt status register and writing a 1 back to the bit position for the interrupt to be cleared.

Independent of the mode, software can read the value in the main counter to see how time has passed between when the interrupt was generated and when it was first serviced.

If Timer 0 is set up to generate a periodic interrupt, the software can check to see how much time remains until the next interrupt by checking the timer value register.

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

A 32-bit timer can be read directly using processors that are capable of 32-bit or 64-bit instructions. However, a 32-bit processor may not be able to directly read 64-bit timer. A race condition comes up if a 32-bit processor reads the 64-bit register using two separate 32-bit reads. The danger is that just after reading one half, the other half rolls over and changes the first half.

If a 32-bit processor needs to access a 64-bit timer, it must first halt the timer before reading both the upper and lower 32-bits of the timer. If a 32-bit processor does not want to halt the timer, it can use the 64-bit timer as a 32-bit timer by setting the TIMERn_32MODE_CNF bit. This causes the timer to behave as a 32-bit timer. The upper 32-bits are always 0.

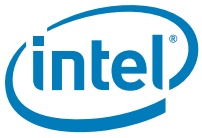
## 5.18 USB UHCI Host Controllers (D29:F0, F1, F2, and F3)

Chipset contains four USB 2.0 full/low-speed host controllers that support the standard Universal Host Controller Interface (UHCI), Revision 1.1. Each UHCI Host Controller (UHC) includes a root hub with two separate USB ports each, for a total of eight USB ports.

- Overcurrent detection on all eight USB ports is supported. The overcurrent inputs are *not* 5 V tolerant, and can be used as GPIs if not needed.
- Chipset's UHCI host controllers are arbitrated differently than standard PCI devices to improve arbitration latency.
- The UHCI controllers use the Analog Front End (AFE) embedded cell that allows support for USB full-speed signaling rates, instead of USB I/O buffers.

### 5.18.1 Data Structures in Main Memory

Section 3.1 - 3.3 of the *Universal Host Controller Interface Specification, Revision 1.1* details the data structures used to communicate control, status, and data between software and Chipset.



### 5.18.2 Data Transfers to/from Main Memory

Section 3.4 of the *Universal Host Controller Interface Specification, Revision 1.1* describes the details on how HCD and Chipset communicate via the Schedule data structures.

### 5.18.3 Data Encoding and Bit Stuffing

Chipset USB employs NRZI data encoding (Non-Return to Zero Inverted) when transmitting packets. Full details on this implementation are given in the *Universal Serial Bus Specification, Revision 2.0*.

### 5.18.4 Bus Protocol

#### 5.18.4.1 Bit Ordering

Bits are sent out onto the bus least significant bit (LSb) first, followed by next LSb, through to the most significant bit (MSb) last.

#### 5.18.4.2 SYNC Field

All packets begin with a synchronization (SYNC) field, which is a coded sequence that generates a maximum edge transition density. The SYNC field appears on the bus as IDLE followed by the binary string “KJKJKJKK,” in its NRZI encoding. It is used by the input circuitry to align incoming data with the local clock and is defined to be 8 bits in length. SYNC serves only as a synchronization mechanism and is not shown in the following packet diagrams. The last two bits in the SYNC field are a marker that is used to identify the first bit of the PID. All subsequent bits in the packet must be indexed from this point.

#### 5.18.4.3 Packet Field Formats

All packets have distinct start and end of packet delimiters. Full details are given in the *Universal Serial Bus Specification, Revision 2.0,* in Section 8.3.1.

#### 5.18.4.4 Address Fields

Function endpoints are addressed using the function address field and the endpoint field. Full details on this are given in the *Universal Serial Bus Specification, Revision 2.0*, in Section 8.3.2.

#### 5.18.4.5 Frame Number Field

The frame number field is an 11-bit field that is incremented by the host on a per frame basis. The frame number field rolls over upon reaching its maximum value of 7FFh, and is sent only for SOF tokens at the start of each frame.

#### 5.18.4.6 Data Field

The data field may range from 0 to 1023 bytes and must be an integral numbers of bytes. Data bits within each byte are shifted out LSB first.

#### 5.18.4.7 Cyclic Redundancy Check (CRC)

CRC is used to protect the all non-PID fields in token and data packets. In this context, these fields are considered to be protected fields. Full details on this are given in the *Universal Serial Bus Specification, Revision 2.0*, in Section 8.3.5.

### 5.18.5 Packet Formats

The USB protocol calls out several packet types: token, data, and handshake packets. Full details on this are given in the *Universal Serial Bus Specification, Revision 2.0*, in section 8.4.

### 5.18.6 USB Interrupts

There are two general groups of USB interrupt sources, those resulting from execution of transactions in the schedule, and those resulting from an Chipset operation error. All transaction-based sources can be masked by software through chipset's Interrupt Enable register. Additionally, individual transfer descriptors can be marked to generate an interrupt on completion.

When Chipset drives an interrupt for USB, it internally drives the PIRQA# pin for USB function #0 and USB function #3, PIRQD# pin for USB function #1, and the PIRQC# pin for USB function #2, until all sources of the interrupt are cleared. In order to accommodate some operating systems, the Interrupt Pin register must contain a different value for each function of this new multi-function device.

#### 5.18.6.1 Transaction-Based Interrupts

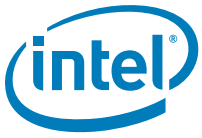
These interrupts are not signaled until after the status for the last complete transaction in the frame has been written back to host memory. This ensures that software can safely process through (Frame List Current Index -1) when it is servicing an interrupt.

**CRC Error / Time-Out**

A CRC/Time-Out error occurs when a packet transmitted from Chipset to a USB device or a packet transmitted from a USB device to Chipset generates a CRC error. Chipset is informed of this event by a time-out from the USB device or by chipset's CRC checker generating an error on reception of the packet. Additionally, a USB bus time-out occurs when USB devices do not respond to a transaction phase within 19-bit times of an EOP. Either of these conditions causes the C_ERR field of the TD to decrement.

When the C_ERR field decrements to 0, the following occurs:

- The Active bit in the TD is cleared



- The Stalled bit in the TD is set
- The CRC/Time-out bit in the TD is set.
- At the end of the frame, the USB Error Interrupt bit is set in the HC status register.

If the CRC/Time out interrupt is enabled in the Interrupt Enable register, a hardware interrupt will be signaled to the system.

### Interrupt on Completion

Transfer Descriptors contain a bit that can be set to cause an interrupt on their completion. The completion of the transaction associated with that block causes the USB Interrupt bit in the HC Status Register to be set at the end of the frame in which the transfer completed. When a TD is encountered with the IOC bit set to 1, the IOC bit in the HC Status register is set to 1 at the end of the frame if the active bit in the TD is set to 0 (even if it was set to 0 when initially read).

If the IOC Enable bit of Interrupt Enable register (bit 2 of I/O offset 04h) is set, a hardware interrupt is signaled to the system. The USB Interrupt bit in the HC status register is set either when the TD completes successfully or because of errors. If the completion is because of errors, the USB Error bit in the HC status register is also set.

### Short Packet Detect

A transfer set is a collection of data which requires more than one USB transaction to completely move the data across the USB. An example might be a large print file which requires numerous TDs in multiple frames to completely transfer the data. Reception of a data packet that is less than the endpoint's Max Packet size during Control, Bulk or Interrupt transfers signals the completion of the transfer set, even if there are active TDs remaining for this transfer set. Setting the SPD bit in a TD indicates to the HC to set the USB Interrupt bit in the HC status register at the end of the frame in which this event occurs. This feature streamlines the processing of input on these transfer types. If the Short Packet Interrupt Enable bit in the Interrupt Enable register is set, a hardware interrupt is signaled to the system at the end of the frame where the event occurred.

#### Serial Bus Babble

When a device transmits on the USB for a time greater than its assigned Max Length, it is said to be babbling. Since isochrony can be destroyed by a babbling device, this error results in the Active bit in the TD being cleared to 0 and the Stalled and Babble bits being set to 1. The C_ERR field is not decremented for a babble. The USB Error Interrupt bit in the HC Status register is set to 1 at the end of the frame. A hardware interrupt is signaled to the system.

If an EOF babble was caused by Chipset (due to incorrect schedule for instance), Chipset forces a bit stuff error followed by an EOP and the start of the next frame.

#### Stalled

This event indicates that a device/endpoint returned a STALL handshake during a transaction or that the transaction ended in an error condition. The TDs Stalled bit is set and the Active bit is cleared. Reception of a STALL does not decrement the error counter. A hardware interrupt is signaled to the system.

#### Data Buffer Error

This event indicates that an overrun of incoming data or a under-run of outgoing data has occurred for this transaction. This would generally be caused by Chipset not being able to access required data buffers in memory within necessary latency requirements. Either of these conditions causes the C_ERR field of the TD to be decremented.

When C_ERR decrements to 0, the Active bit in the TD is cleared, the Stalled bit is set, the USB Error Interrupt bit in the HC Status register is set to 1 at the end of the frame and a hardware interrupt is signaled to the system.

#### Bit Stuff Error

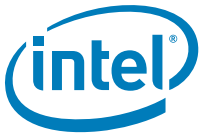
A bit stuff error results from the detection of a sequence of more that six 1s in a row within the incoming data stream. This causes the C_ERR field of the TD to be decremented. When the C_ERR field decrements to 0, the Active bit in the TD is cleared to 0, the Stalled bit is set to 1, the USB Error Interrupt bit in the HC Status register is set to 1 at the end of the frame and a hardware interrupt is signaled to the system.

### 5.18.6.2 Non-Transaction Based Interrupts

If an Chipset process error or system error occur, Chipset halts and immediately issues a hardware interrupt to the system.

#### Resume Received

This event indicates that Chipset received a RESUME signal from a device on the USB bus during a global suspend. If this interrupt is enabled in the Interrupt Enable register, a hardware interrupt is signaled to the system allowing the USB to be brought out of the suspend state and returned to normal operation.



### Chipset Process Error

The HC monitors certain critical fields during operation to ensure that it does not process corrupted data structures. These include checking for a valid PID and verifying that the MaxLength field is less than 1280. If it detects a condition that would indicate that it is processing corrupted data structures, it immediately halts processing, sets the HC Process Error bit in the HC Status register and signals a hardware interrupt to the system.

This interrupt cannot be disabled through the Interrupt Enable register.

### Host System Error

Chipset sets this bit to 1 when a Parity error, Master Abort, or Target Abort occur. When this error occurs, Chipset clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs. This interrupt cannot be disabled through the Interrupt Enable register.

## 5.18.7 USB Power Management

The Host controller can be put into a suspended state and its power can be removed. This requires that certain bits of information are retained in the resume power plane of Chipset so that a device on a port may wake the system. Such a device may be a fax-modem, which will wake up the machine to receive a fax or take a voice message. The settings of the following bits in I/O space will be maintained when Chipset enters the S3, S4, or S5 states.

**Table 5-72.Bits Maintained in Low Power States**

| Register | Offset | Bit | Description |
|---|---|---|---|
| Command | 00h | 3 | Enter Global Suspend Mode (EGSM) |
| Status | 02h | 2 | Resume Detect |
| Port Status and Control | 10h & 12h | 2 | Port Enabled/Disabled |
| | | 6 | Resume Detect |
| | | 8 | Low-speed Device Attached |
| | | 12 | Suspend |

When Chipset detects a resume event on any of its ports, it sets the corresponding USB_STS bit in ACPI space. If USB is enabled as a wake/break event, the system wakes up and an SCI generated.

## 5.18.8 USB Legacy Keyboard Operation

When a USB keyboard is plugged into the system, and a standard keyboard is not, the system may not boot, and MS-DOS legacy software will not run, because the keyboard will not be identified. Chipset implements a series of trapping operations which will snoop accesses that go to the keyboard controller, and put the expected data from the USB keyboard into the keyboard controller.

***Note:*** The scheme described below assumes that the keyboard controller (8042 or equivalent) is on the LPC bus.

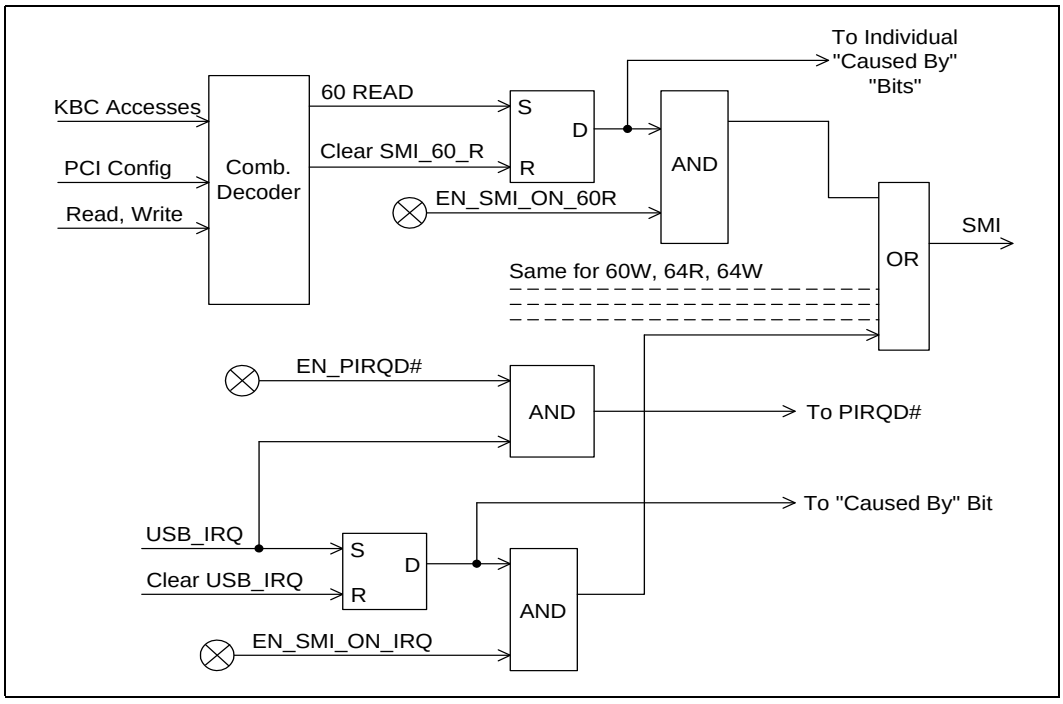
This legacy operation is performed through SMM space. Figure 5-14 shows the Enable and Status path. The latched SMI source (60R, 60W, 64R, 64W) is available in the Status Register. Because the enable is after the latch, it is possible to check for other events that didn't necessarily cause an SMI. It is the software's responsibility to logically AND the value with the appropriate enable bits.

Note also that the SMI is generated before the PCI cycle completes (e.g., before TRDY# goes active) to ensure that the processor doesn't complete the cycle before the SMI is observed. This method is used on MPIIX and has been validated.

The logic also needs to block the accesses to the 8042. If there is an external 8042, then this is simply accomplished by not activating the 8042 CS. This is simply done by logically ANDing the four enables (60R, 60W, 64R, 64W) with the 4 types of accesses to determine if 8042CS should go active. An additional term is required for the “pass-through” case.

The state table for the diagram is shown in Table 5-73.

**Figure 5-14. USB Legacy Keyboard Flow Diagram**



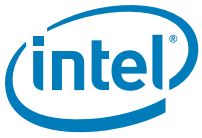


**Table 5-73.USB Legacy Keyboard State Transitions (Sheet 1 of 2)**

| Current State | Action | Data Value | Next State | Comment |
|---|---|---|---|---|
| IDLE | 64h / Write | D1h | GateState1 | Standard D1 command. Cycle passed through to 8042. SMI# doesn't go active. PSTATE (offset C0, bit 6) goes to 1. |
| IDLE | 64h / Write | Not D1h | IDLE | Bit 3 in Config Register determines if cycle passed through to 8042 and if SMI# generated. |
| IDLE | 64h / Read | N/A | IDLE | Bit 2 in Config Register determines if cycle passed through to 8042 and if SMI# generated. |
| IDLE | 60h / Write | Don't Care | IDLE | Bit 1 in Config Register determines if cycle passed through to 8042 and if SMI# generated. |
| IDLE | 60h / Read | N/A | IDLE | Bit 0 in Config Register determines if cycle passed through to 8042 and if SMI# generated. |
| GateState1 | 60h / Write | XXh | GateState2 | Cycle passed through to 8042, even if trap enabled in Bit 1 in Config Register. No SMI# generated. PSTATE remains 1. If data value is not DFh or DDh then the 8042 may chose to ignore it. |
| GateState1 | 64h / Write | D1h | GateState1 | Cycle passed through to 8042, even if trap enabled via Bit 3 in Config Register. No SMI# generated. PSTATE remains 1. Stay in GateState1 because this is part of the double-trigger sequence. |
| GateState1 | 64h / Write | Not D1h | ILDE | Bit 3 in Config space determines if cycle passed through to 8042 and if SMI# generated. PSTATE goes to 0. If Bit 7 in Config Register is set, then SMI# should be generated. |
| GateState1 | 60h / Read | N/A | IDLE | This is an invalid sequence. Bit 0 in Config Register determines if cycle passed through to 8042 and if SMI# generated. PSTATE goes to 0. If Bit 7 in Config Register is set, then SMI# should be generated. |
| GateState1 | 64h / Read | N/A | GateState1 | Just stay in same state. Generate an SMI# if enabled in Bit 2 of Config Register. PSTATE remains 1. |
| GateState2 | 64 / Write | FFh | IDLE | Standard end of sequence. Cycle passed through to 8042. PSTATE goes to 0. Bit 7 in Config Space determines if SMI# should be generated. |
| GateState2 | 64h / Write | Not FFh | IDLE | Improper end of sequence. Bit 3 in Config Register determines if cycle passed through to 8042 and if SMI# generated. PSTATE goes to 0. If Bit 7 in Config Register is set, then SMI# should be generated. |

**Table 5-73.USB Legacy Keyboard State Transitions (Sheet 2 of 2)**

| Current State | Action | Data Value | Next State | Comment |
|---|---|---|---|---|
| GateState2 | 64h / Read | N/A | GateState2 | Just stay in same state. Generate an SMI# if enabled in Bit 2 of Config Register. PSTATE remains 1. |
| GateState2 | 60h / Write | XXh | IDLE | Improper end of sequence. Bit 1 in Config Register determines if cycle passed through to 8042 and if SMI# generated. PSTATE goes to 0. If Bit 7 in Config Register is set, then SMI# should be generated. |
| GateState2 | 60h / Read | N/A | IDLE | Improper end of sequence. Bit 0 in Config Register determines if cycle passed through to 8042 and if SMI# generated. PSTATE goes to 0. If Bit 7 in Config Register is set, then SMI# should be generated. |

## 5.19 USB EHCI Host Controller (D29:F7)

Chipset contains an Enhanced Host Controller Interface (EHCI) host controller which supports up to eight USB 2.0 high-speed root ports. USB 2.0 allows data transfers up to 480 Mb/s using the same pins as the eight USB full-speed/low-speed ports. Chipset contains port-routing logic that determines whether a USB port is controlled by one of the UHCI controllers or by the EHCI controller. USB 2.0 based Debug Port is also implemented in Chipset.

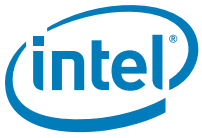
A summary of the key architectural differences between the USB UHCI host controllers and the EHCI host controller are shown in Table 5-74.

**Table 5-74.UHCI vs. EHCI**

| Parameter | USB UHCI | USB EHCI |
|---|---|---|
| Accessible by | I/O space | Memory Space |
| Memory Data Structure | Single linked list | Separated in to Periodic and Asynchronous lists |
| Differential Signaling Voltage | 3.3 V | 400 mV |
| Ports per Controller | 2 | 8 |

### 5.19.1 EHC Initialization

The following descriptions step through the expected Chipset Enhanced Host Controller (EHC) initialization sequence in chronological order, beginning with a complete power cycle in which the suspend well and core well have been off.



#### 5.19.1.1 BIOS Initialization

BIOS performs a number of platform customization steps after the core well has powered up. Contact your Intel Field Representative for additional Chipset BIOS information.

#### 5.19.1.2 Driver Initialization

See Chapter 4 of the *Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0.

#### 5.19.1.3 EHC Resets

In addition to the standard Chipset hardware resets, portions of the EHC are reset by the HCRESET bit and the transition from the $D3_{HOT}$ device power management state to the D0 state. The effects of each of these resets are:

| Reset | Does Reset | Does not Reset | Comments |
|---|---|---|---|
| HCRESET bit set. | Memory space registers except Structural Parameters (which is written by BIOS). | Configuration registers. | The HCRESET must only affect registers that the EHCI driver controls. PCI Configuration space and BIOS-programmed parameters can not be reset. |
| Software writes the Device Power State from $D3_{HOT}$ (11b) to D0 (00b). | Core well registers (except BIOS-programmed registers). | Suspend well registers; BIOS-programmed core well registers. | The D3-to-D0 transition must not cause wake information (suspend well) to be lost. It also must not clear BIOS-programmed registers because BIOS may not be invoked following the D3-to-D0 transition. |

If the detailed register descriptions give exceptions to these rules, those exceptions override these rules. This summary is provided to help explain the reasons for the reset policies.

### 5.19.2 Data Structures in Main Memory

See Section 3 and Appendix B of the *Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0 for details.

### 5.19.3 USB 2.0 Enhanced Host Controller DMA

Chipset USB 2.0 EHC implements three sources of USB packets. They are, in order of priority on USB during each microframe:

1. The USB 2.0 Debug Port (see Section 5.19.10),
2. The Periodic DMA engine, and
3. The Asynchronous DMA engine. Chipset always performs any currently-pending debug port transaction at the beginning of a microframe, followed by any pending

periodic traffic for the current microframe. If there is time left in the microframe, then the EHC performs any pending asynchronous traffic until the end of the microframe (EOF1). Note that the debug port traffic is only presented on one port (Port #0), while the other ports are idle during this time.

### 5.19.4 Data Encoding and Bit Stuffing

See Chapter 8 of the *Universal Serial Bus Specification, Revision 2.0.*

### 5.19.5 Packet Formats

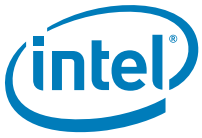
See Chapter 8 of the *Universal Serial Bus Specification, Revision 2.0*.

Chipset EHCI allows entrance to USB test modes, as defined in the USB 2.0 specification, including Test J, Test Packet, etc. However, note that Chipset Test Packet test mode interpacket gap timing may not meet the USB 2.0 specification.

### 5.19.6 USB 2.0 Interrupts and Error Conditions

Section 4 of the *Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0 goes into detail on the EHC interrupts and the error conditions that cause them. All error conditions that the EHC detects can be reported through the EHCI Interrupt status bits. Only Chipset-specific interrupt and error-reporting behavior is documented in this section. The EHCI Interrupts Section must be read first, followed by this section of the datasheet to fully comprehend the EHC interrupt and error-reporting functionality.

- Based on the EHC's Buffer sizes and buffer management policies, the Data Buffer Error can not occur on Chipset.
- Master Abort and Target Abort responses from hub interface on EHC-initiated read packets will be treated as Fatal Host Errors. The EHC halts when these conditions are encountered.
- Chipset may assert the interrupts which are based on the interrupt threshold as soon as the status for the last complete transaction in the interrupt interval has been posted in the internal write buffers. The requirement in the *Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0 (that the status is written to memory) is met internally, even though the write may not be seen on DMI before the interrupt is asserted.
- Since Chipset supports the 1024-element Frame List size, the Frame List Rollover interrupt occurs every 1024 milliseconds.
- Chipset delivers interrupts using PIRQH#.
- Chipset does not modify the CERR count on an Interrupt IN when the "Do Complete-Split" execution criteria are not met.
- For complete-split transactions in the Periodic list, the "Missed Microframe" bit does not get set on a control-structure-fetch that fails the late-start test. If subsequent



accesses to that control structure do not fail the late-start test, then the “Missed Microframe” bit will get set and written back.

#### 5.19.6.1 Aborts on USB 2.0-Initiated Memory Reads

If a read initiated by the EHC is aborted, the EHC treats it as a fatal host error. The following actions are taken when this occurs:

- The Host System Error status bit is set
- The DMA engines are halted after completing up to one more transaction on the USB interface
- If enabled (by the Host System Error Enable), then an interrupt is generated
- If the status is Master Abort, then the Received Master Abort bit in configuration space is set
- If the status is Target Abort, then the Received Target Abort bit in configuration space is set
- If enabled (by the SERR Enable bit in the function’s configuration space), then the Signaled System Error bit in configuration bit is set.

### 5.19.7 USB 2.0 Power Management

#### 5.19.7.1 Pause Feature

This feature allows platforms (especially Netbook systems) to dynamically enter low-power states during brief periods when the system is idle (i.e., between keystrokes). This is useful for enabling power management features like Intel SpeedStep technology in Chipset Netbook Only. The policies for entering these states typically are based on the recent history of system bus activity to incrementally enter deeper power management states. Normally, when the EHC is enabled, it regularly accesses main memory while traversing the DMA schedules looking for work to do; this activity is viewed by the power management software as a non-idle system, thus preventing the power managed states to be entered. Suspending all of the enabled ports can prevent the memory accesses from occurring, but there is an inherent latency overhead with entering and exiting the suspended state on the USB ports that makes this unacceptable for the purpose of dynamic power management. As a result, the EHCI software drivers are allowed to pause the EHC’s DMA engines when it knows that the traffic patterns of the attached devices can afford the delay. The pause only prevents the EHC from generating memory accesses; the SOF packets continue to be generated on the USB ports (unlike the suspended state).

#### 5.19.7.2 Suspend Feature

The *Enhanced Host Controller Interface (EHCI) For Universal Serial Bus Specification*, Section 4.3 describes the details of Port Suspend and Resume.

### 5.19.7.3 ACPI Device States

The USB 2.0 function only supports the D0 and D3 PCI Power Management states. Notes regarding Chipset implementation of the Device States:

1. The EHC hardware does not inherently consume any more power when it is in the D0 state than it does in the D3 state. However, software is required to suspend or disable all ports prior to entering the D3 state such that the maximum power consumption is reduced.
2. In the D0 state, all implemented EHC features are enabled.
3. In the D3 state, accesses to the EHC memory-mapped I/O range will master abort. Note that, since the Debug Port uses the same memory range, the Debug Port is only operational when the EHC is in the D0 state.
4. In the D3 state, the EHC interrupt must not assert for any reason. The internal PME# signal is used to signal wake events, etc.
5. When the Device Power State field is written to D0 from D3, an internal reset is generated. See section EHC Resets for general rules on the effects of this reset.
6. Attempts to write any other value into the Device Power State field other than 00b (D0 state) and 11b (D3 state) will complete normally without changing the current value in this field.

### 5.19.7.4 ACPI System States

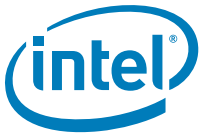
The EHC behavior as it relates to other power management states in the system is summarized in the following list:

- The System is always in the S0 state when the EHC is in the D0 state. However, when the EHC is in the D3 state, the system may be in any power management state (including S0).
- When in D0, the Pause feature (See Section 5.19.7.1) enables dynamic processor low-power states to be entered.
- The PLL in the EHC is disabled when entering the $S3_{HOT}$ state (48 MHz clock stops), or the $S3_{COLD}$/S4/S5 states (core power turns off).
- All core well logic is reset in the S3/S4/S5 states.

### 5.19.7.5 Netbook Only Considerations

Chipset USB 2.0 implementation does not behave differently in the Netbook configurations versus the Nettop configurations. However, some features may be especially useful for the Netbook configurations.

- If a system (e.g., Netbook) does not implement all eight USB 2.0 ports, Chipset provides mechanisms for changing the structural parameters of the EHC and hiding unused UHCI controllers. See Chipset *BIOS Specification* for information on how BIOS should configure Chipset.



- Netbook systems may want to minimize the conditions that will wake the system. Chipset implements the “Wake Enable” bits in the Port Status and Control registers, as specified in the EHCI spec, for this purpose.
- Netbook systems may want to cut suspend well power to some or all USB ports when in a low-power state. Chipset implements the optional Port Wake Capability Register in the EHC Configuration Space for this platform-specific information to be communicated to software.

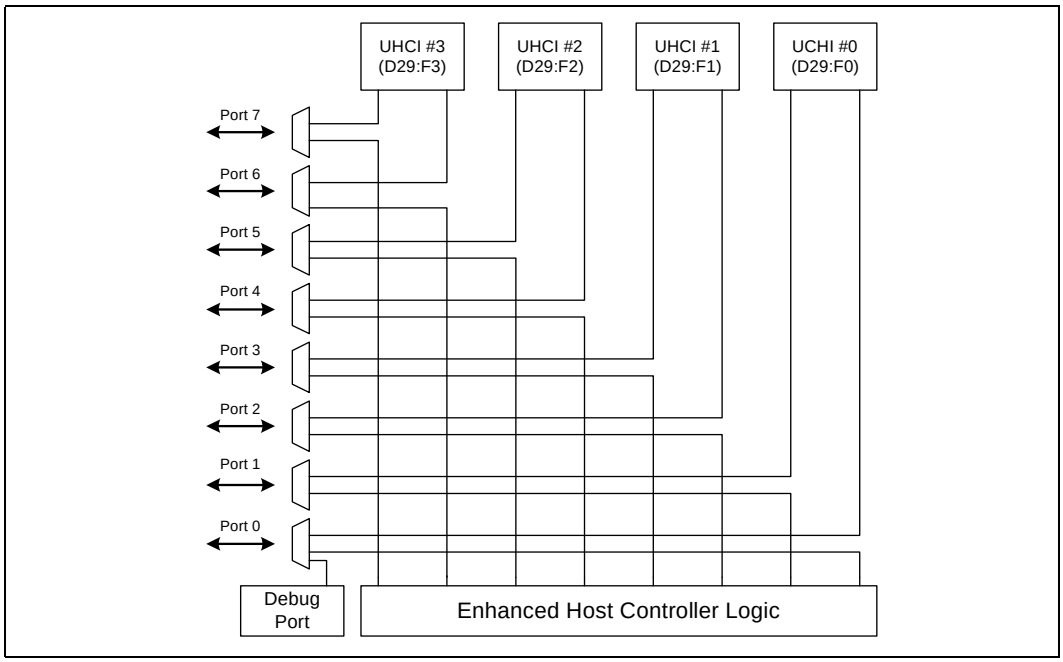
## 5.19.8 Interaction with UHCI Host Controllers

The Enhanced Host controller shares the eight USB ports with four UHCI Host controllers in Chipset. The UHC at D29:F0 shares ports 0 and 1; the UHC at D29:F1 shares ports 2 and 3; the UHC at D29:F2 shares ports 4 and 5; and the UHC at D29:F3 shares ports 6 and 7 with the EHC. There is very little interaction between the Enhanced and the UHCI controllers other than the muxing control which is provided as part of the EHC.Figure 5-15 shows the USB Port Connections at a conceptual level.

### 5.19.8.1 Port-Routing Logic

Integrated into the EHC functionality is port-routing logic, which performs the muxing between the UHCI and EHCI host controllers. Chipset conceptually implements this logic as described in Section 4.2 of the *Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0. If a device is connected that is not capable of USB 2.0’s high-speed signaling protocol or if the EHCI software drivers are not present as indicated by the Configured Flag, then the UHCI controller owns the port. Owning the port means that the differential output is driven by the owner and the input stream is only visible to the owner. The host controller that is not the owner of the port internally sees a disconnected port.

**Figure 5-15. Chipset-USB Port Connections**




Note that the port-routing logic is the only block of logic within Chipset that observes the physical (real) connect/disconnect information. The port status logic inside each of the host controllers observes the electrical connect/disconnect information that is generated by the port-routing logic.

Only the differential signal pairs are multiplexed/demultiplexed between the UHCI and EHCI host controllers. The other USB functional signals are handled as follows:

- The Overcurrent inputs (OC[7:0]#) are directly routed to both controllers. An overcurrent event is recorded in both controllers' status registers.

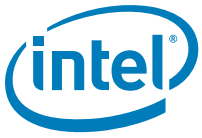
The Port-Routing logic is implemented in the Suspend power well so that re-enumeration and
re-mapping of the USB ports is not required following entering and exiting a system sleep state in which the core power is turned off.

Chipset also allows the USB Debug Port traffic to be routed in and out of Port #0. When in this mode, the Enhanced Host controller is the owner of Port #0.

### 5.19.8.2 Device Connects

The *Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0 describes the details of handling Device Connects in Section 4.2. There are four general scenarios that are summarized below.

1. Configure Flag = 0 and a full-speed/low-speed-only Device is connected



— In this case, the UHC is the owner of the port both before and after the connect occurs. The EHC (except for the port-routing logic) does not see the connect occur. The UHCI driver handles the connection and initialization process.

2. Configure Flag = 0 and a high-speed-capable Device is connected

   — In this case, the UHC is the owner of the port both before and after the connect occurs. The EHC (except for the port-routing logic) not see the connect occur. The UHCI driver handles the connection and initialization process. Since the UHC does not perform the high-speed chirp handshake, the device operates in compatible mode.

3. Configure Flag = 1 and a full-speed/low-speed-only Device is connected

   — In this case, the EHC is the owner of the port before the connect occurs. The EHCI driver handles the connection and performs the port reset. After the reset process completes, the EHC hardware has cleared (not set) the Port Enable bit in the EHC's PORTSC register. The EHCI driver then writes a 1 to the Port Owner bit in the same register, causing the UHC to see a connect event and the EHC to see an "electrical" disconnect event. The UHCI driver and hardware handle the connection and initialization process from that point on. The EHCI driver and hardware handle the perceived disconnect.

4. Configure Flag = 1 and a high-speed-capable Device is connected

   — In this case, the EHC is the owner of the port before, and remains the owner after, the connect occurs. The EHCI driver handles the connection and performs the port reset. After the reset process completes, the EHC hardware has set the Port Enable bit in the EHC's PORTSC register. The port is functional at this point. The UHC continues to see an unconnected port.

#### 5.19.8.3 Device Disconnects

The *Enhanced Host Controller Interface Specification for Universal Serial Bus,* Revision 1.0 describes the details of handling Device Connects in Section 4.2. There are three general scenarios that are summarized below.

1. Configure Flag = 0 and the device is disconnected

   — In this case, the UHC is the owner of the port both before and after the disconnect occurs. The EHC (except for the port-routing logic) not see a device attached. The UHCI driver handles disconnection process.

2. Configure Flag = 1 and a full-speed/low-speed-capable Device is disconnected

   — In this case, the UHC is the owner of the port before the disconnect occurs. The disconnect is reported by the UHC and serviced by the associated UHCI driver. The port-routing logic in the EHC cluster forces the Port Owner bit to 0, indicating that the EHC owns the unconnected port.

3. Configure Flag = 1 and a high-speed-capable Device is disconnected

   — In this case, the EHC is the owner of the port before, and remains the owner after, the disconnect occurs. The EHCI hardware and driver handle the disconnection process. The UHC does not see a device attached.

#### 5.19.8.4 Effect of Resets on Port-Routing Logic

As mentioned above, the Port Routing logic is implemented in the suspend power well so that remuneration and re-mapping of the USB ports is not required following entering and exiting a system sleep state in which the core power is turned off.

| Reset Event | Effect on Configure Flag | Effect on Port Owner Bits |
|---|---|---|
| Suspend Well Reset | cleared (0) | set (1) |
| Core Well Reset | no effect | no effect |
| D3-to-D0 Reset | no effect | no effect |
| HCRESET | cleared (0) | set (1) |

### 5.19.9 USB 2.0 Legacy Keyboard Operation

Chipset must support the possibility of a keyboard downstream from either a full-speed/low-speed or a high-speed port. The description of the legacy keyboard support is unchanged from USB 1.1 (See Section 5.18.8).

The EHC provides the basic ability to generate SMIs on an interrupt event, along with more sophisticated control of the generation of SMIs.

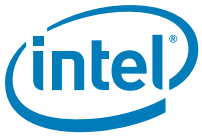
### 5.19.10 USB 2.0 Based Debug Port

Chipset supports the elimination of the legacy COM ports by providing the ability for new debugger software to interact with devices on a USB 2.0 port.

High-level restrictions and features are:

- Operational before USB 2.0 drivers are loaded.
- Functions even when the port is disabled.
- Works even though non-configured port is default-routed to the UHCI. Note that the Debug Port can not be used to debug an issue that requires a full-speed/low-speed device on Port #0 using the UHCI drivers.
- Allows normal system USB 2.0 traffic in a system that may only have one USB port.
- Debug Port device (DPD) must be high-speed capable and connect directly to Port #0 on Chipset systems (e.g., the DPD cannot be connected to Port #0 thru a hub).
- Debug Port FIFO always makes forward progress (a bad status on USB is simply presented back to software).
- The Debug Port FIFO is only given one USB access per microframe.

The Debug port facilitates operating system and device driver debug. It allows the software to communicate with an external console using a USB 2.0 connection. Because the interface to this link does not go through the normal USB 2.0 stack, it allows communication with the external console during cases where the operating system is not loaded, the USB 2.0 software is broken, or where the USB 2.0 software is being debugged. Specific features of this implementation of a debug port are:



- Only works with an external USB 2.0 debug device (console)
- Implemented for a specific port on the host controller
- Operational anytime the port is not suspended AND the host controller is in D0 power state.
- Capability is interrupted when port is driving USB RESET

### 5.19.10.1 Theory of Operation

There are two operational modes for the USB debug port:

1. Mode 1 is when the USB port is in a disabled state from the viewpoint of a standard host controller driver. In Mode 1, the Debug Port controller is required to generate a “keepalive” packets less than 2 ms apart to keep the attached debug device from suspending. The keepalive packet should be a standalone 32-bit SYNC field.
2. Mode 2 is when the host controller is running (i.e., host controller’s *Run/Stop#* bit is 1). In Mode 2, the normal transmission of SOF packets will keep the debug device from suspending.

#### Behavioral Rules

1. In both modes 1 and 2, the Debug Port controller must check for software requested debug transactions at least every 125 microseconds.
2. If the debug port is enabled by the debug driver, and the standard host controller driver resets the USB port, USB debug transactions are held off for the duration of the reset and until after the first SOF is sent.
3. If the standard host controller driver suspends the USB port, then USB debug transactions are held off for the duration of the suspend/resume sequence and until after the first SOF is sent.
4. The ENABLED_CNT bit in the debug register space is independent of the similar port control bit in the associated Port Status and Control register.

Table 5-75 shows the debug port behavior related to the state of bits in the debug registers as well as bits in the associated Port Status and Control register.

**Table 5-75.Debug Port Behavior (Sheet 1 of 2)**

| OWNER_CNT | ENABLED_CT | Port Enable | Run / Stop | Suspend | Debug Port Behavior |
|---|---|---|---|---|---|
| 0 | X | X | X | X | Debug port is not being used. Normal operation. |
| 1 | 0 | X | X | X | Debug port is not being used. Normal operation. |
| 1 | 1 | 0 | 0 | X | Debug port in Mode 1. SYNC keepalives sent plus debug traffic |

**Table 5-75.Debug Port Behavior (Sheet 2 of 2)**

| OWNER_CNT | ENABLED_CT | Port Enable | Run / Stop | Suspend | Debug Port Behavior |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | X | Debug port in Mode 2. SOF (and only SOF) is sent as keepalive. Debug traffic is also sent. Note that no other normal traffic is sent out this port, because the port is not enabled. |
| 1 | 1 | 1 | 0 | 0 | Invalid. Host controller driver should not put the controller into this state (enabled, not running and not suspended). |
| 1 | 1 | 1 | 0 | 1 | Port is suspended. No debug traffic sent. |
| 1 | 1 | 1 | 1 | 0 | Debug port in Mode 2. Debug traffic is interspersed with normal traffic. |
| 1 | 1 | 1 | 1 | 1 | Port is suspended. No debug traffic sent. |

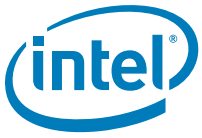
#### 5.19.10.1.1 OUT Transactions

An Out transaction sends data to the debug device. It can occur only when the following are true:

- The debug port is enabled
- The debug software sets the GO_CNT bit
- The WRITE_READ#_CNT bit is set

The sequence of the transaction is:

1. Software sets the appropriate values in the following bits:
   - — USB_ADDRESS_CNF
   - — USB_ENDPOINT_CNF
   - — DATA_BUFFER[63:0]
   - — TOKEN_PID_CNT[7:0]
   - — SEND_PID_CNT[15:8]
   - — DATA_LEN_CNT
   - — WRITE_READ#_CNT (note: this will always be 1 for OUT transactions)
   - — GO_CNT (note: this will always be 1 to initiate the transaction)
2. The debug port controller sends a token packet consisting of:
   - — SYNC
   - — TOKEN_PID_CNT field



- — USB_ADDRESS_CNT field
- — USB_ENDPOINT_CNT field
- — 5-bit CRC field

3. After sending the token packet, the debug port controller sends a data packet consisting of:
   - — SYNC
   - — SEND_PID_CNT field
   - — The number of data bytes indicated in DATA_LEN_CNT from the DATA_BUFFER
   - — 16-bit CRC

NOTE: A DATA_LEN_CNT value of 0 is valid in which case no data bytes would be included in the packet.

4. After sending the data packet, the controller waits for a handshake response from the debug device.

- If a handshake is received, the debug port controller:
  - — a. Places the received PID in the RECEIVED_PID_STS field
  - — b. Resets the ERROR_GOOD#_STS bit
  - — c. Sets the DONE_STS bit
- If no handshake PID is received, the debug port controller:
  - — a. Sets the EXCEPTION_STS field to 001b
  - — b. Sets the ERROR_GOOD#_STS bit
  - — c. Sets the DONE_STS bit

#### 5.19.10.1.2 IN Transactions

An IN transaction receives data from the debug device. It can occur only when the following are true:

- The debug port is enabled
- The debug software sets the GO_CNT bit
- The WRITE_READ#_CNT bit is reset

The sequence of the transaction is:

1. Software sets the appropriate values in the following bits:
   - — USB_ADDRESS_CNF
   - — USB_ENDPOINT_CNF
   - — TOKEN_PID_CNT[7:0]
   - — DATA_LEN_CNT
   - — WRITE_READ#_CNT (note: this will always be 0 for IN transactions)

— GO_CNT (note: this will always be 1 to initiate the transaction)

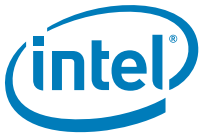
2. The debug port controller sends a token packet consisting of:
   — SYNC
   — TOKEN_PID_CNT field
   — USB_ADDRESS_CNT field
   — USB_ENDPOINT_CNT field
   — 5-bit CRC field.
3. After sending the token packet, the debug port controller waits for a response from the debug device.
   If a response is received:
   — The received PID is placed into the RECEIVED_PID_STS field
   — Any subsequent bytes are placed into the DATA_BUFFER
   — The DATA_LEN_CNT field is updated to show the number of bytes that were received after the PID.
4. If valid packet was received from the device that was one byte in length (indicating it was a handshake packet), then the debug port controller:
   — Resets the ERROR_GOOD#_STS bit
   — Sets the DONE_STS bit
5. If valid packet was received from the device that was more than one byte in length (indicating it was a data packet), then the debug port controller:
   — Transmits an ACK handshake packet
   — Resets the ERROR_GOOD#_STS bit
   — Sets the DONE_STS bit
6. If no valid packet is received, then the debug port controller:
   — Sets the EXCEPTION_STS field to 001b
   — Sets the ERROR_GOOD#_STS bit
   — Sets the DONE_STS bit.

#### 5.19.10.1.3 Debug Software

**Enabling the Debug Port**

There are two mutually exclusive conditions that debug software must address as part of its startup processing:

- The EHCI has been initialized by system software
- The EHCI has not been initialized by system software



Debug software can determine the current 'initialized' state of the EHCI by examining the Configure Flag in the EHCI USB 2.0 Command Register. If this flag is set, then system software has initialized the EHCI. Otherwise the EHCI should not be considered initialized. Debug software will initialize the debug port registers depending on the state the EHCI. However, before this can be accomplished, debug software must determine which root USB port is designated as the debug port.

### Determining the Debug Port

Debug software can easily determine which USB root port has been designated as the debug port by examining bits 20:23 of the EHCI Host Controller Structural Parameters register. This 4-bit field represents the numeric value assigned to the debug port (i.e., 0001=port 0).

### Debug Software Startup with Non-Initialized EHCI

Debug software can attempt to use the debug port if after setting the OWNER_CNT bit, the Current Connect Status bit in the appropriate (See Determining the Debug Port) PORTSC register is set. If the Current Connect Status bit is not set, then debug software may choose to terminate or it may choose to wait until a device is connected.

If a device is connected to the port, then debug software must reset/enable the port. Debug software does this by setting and then clearing the Port Reset bit the PORTSC register. To ensure a successful reset, debug software should wait at least 50 ms before clearing the Port Reset bit. Due to possible delays, this bit may not change to 0 immediately; reset is complete when this bit reads as 0. Software must not continue until this bit reads 0.

If a high-speed device is attached, the EHCI will automatically set the Port Enabled/Disabled bit in the PORTSC register and the debug software can proceed. Debug software should set the ENABLED_CNT bit in the Debug Port Control/Status register, and then reset (clear) the Port Enabled/Disabled bit in the PORTSC register (so that the system host controller driver does not see an enabled port when it is first loaded).

### Debug Software Startup with Initialized EHCI

Debug software can attempt to use the debug port if the Current Connect Status bit in the appropriate (See Determining the Debug Port) PORTSC register is set. If the Current Connect Status bit is not set, then debug software may choose to terminate or it may choose to wait until a device is connected.

If a device is connected, then debug software must set the OWNER_CNT bit and then the ENABLED_CNT bit in the Debug Port Control/Status register.

### Determining Debug Peripheral Presence

After enabling the debug port functionality, debug software can determine if a debug peripheral is attached by attempting to send data to the debug peripheral. If all attempts result in an error (Exception bits in the Debug Port Control/Status register

indicates a Transaction Error), then the attached device is not a debug peripheral. If the debug port peripheral is not present, then debug software may choose to terminate or it may choose to wait until a debug peripheral is connected.

## 5.20 SMBus Controller (D31:F3)

Chipset provides an System Management Bus (SMBus) 2.0 host controller as well as an SMBus Slave Interface. The host controller provides a mechanism for the processor to initiate communications with SMBus peripherals (slaves). Chipset is also capable of operating in a mode in which it can communicate with $I^2C$ compatible devices.

Chipset can perform SMBus messages with either packet error checking (PEC) enabled or disabled. The actual PEC calculation and checking is performed in hardware by Chipset.

The Slave Interface allows an external master to read from or write to Chipset. 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. Chipset's internal host controller cannot access chipset's internal Slave Interface.

Chipset SMBus logic exists in Device 31:Function 3 configuration space, and consists of a transmit data path, and host controller. The transmit data path provides the data flow logic needed to implement the seven different SMBus command protocols and is controlled by the host controller. Chipset SMBus controller logic is clocked by RTC clock.

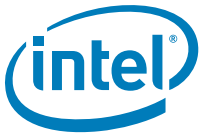
The SMBus Address Resolution Protocol (ARP) is supported by using the existing host controller commands through software, except for the new Host Notify command (which is actually a received message).

The programming model of the host controller is combined into two portions: a PCI configuration portion, and a system I/O mapped portion. All static configuration, such as the I/O base address, is done via the PCI configuration space. Real-time programming of the Host interface is done in system I/O space.

Chipset 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 (Device 31:Function 3:Offset 06h:bit 15) is set. If bit 6 and bit 8 of the PCI Command Register (Device 31:Function 3:Offset 04h) are set, an SERR# is generated and the signaled SERR# bit in the PCI Status Register (bit 14) is set.

### 5.20.1 Host Controller

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



The host controller supports 8 command protocols of the SMBus interface (see *System Management Bus (SMBus) Specification, Version 2.0*): Quick Command, Send Byte, Receive Byte, Write Byte/Word, Read Byte/Word, Process Call, Block Read/Write, Block Write–Block Read Process Call, and Host Notify.

The SMBus host controller requires that the various data and command fields be setup for the type of command to be sent. When software sets the START bit, the SMBus Host controller performs the requested transaction, and interrupts the processor (or generates an SMI#) when the transaction is completed. Once a START command has been issued, the values of the "active registers" (Host Control, Host Command, Transmit Slave Address, Data 0, Data 1) should not be changed or read until the interrupt status bit (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.

Using the SMB host controller to send commands to chipset's SMB slave port is supported. Chipset supports the *System Management Bus (SMBus) Specification, Version 2.0*. Slave functionality, including the Host Notify protocol, is available on the SMBus pins. The SMLink and SMBus signals should not be tied together externally.

#### 5.20.1.1 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.

**Quick Command**

When programmed for a Quick Command, the Transmit Slave Address Register is sent. The PEC byte is not appended to the Quick Protocol. Software should force the PEC_EN bit to 0 when performing the Quick Command. Software must force the I2C_EN bit to 0 when running this command. See section 5.5.1 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

### Send Byte / Receive Byte

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

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

### Write Byte / Word

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

### Read Byte / Word

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

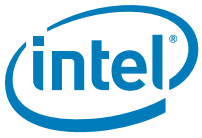
When programmed for the read byte/word command, the Transmit Slave Address and Device Command Registers are sent. Data is received into the DATA0 on the read byte, and the DAT0 and DATA1 registers on the read word. See section 5.5.5 of the *System Management Bus (SMBus) Specification,* Version 2.0 for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

### Process Call

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

When programmed for the Process Call command, Chipset transmits the Transmit Slave Address, Host Command, DATA0 and DATA1 registers. Data received from the device is stored in the DATA0 and DATA1 registers. The Process Call command with I2C_EN set and the PEC_EN bit set produces undefined results. Software must force either I2C_EN or PEC_EN to 0 when running this command. See section 5.5.6 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

***Note:*** For process call command, the value written into bit 0 of the Transmit Slave Address Register (SMB I/O register, offset 04h) needs to be 0.



***Note:*** If the I2C_EN bit is set, the protocol sequence changes slightly: the Command Code (bits 18:11 in the bit sequence) are not sent - as a result, the slave will not acknowledge (bit 19 in the sequence).

### Block Read / Write

Chipset 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 Chipset, 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 Chipset as software will end the transfer after all bytes it cares about have been sent or received.

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

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

When programmed for a block write command, the Transmit Slave Address, Device Command, and Data0 (count) registers are sent. Data is then sent from the Block Data Byte register; the total data sent being the value stored in the Data0 Register. On block read commands, the first byte received is stored in the Data0 register, and the remaining bytes are stored in the Block Data Byte register. See section 5.5.7 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

***Note:*** For Block Write, if the I2C_EN bit is set, the format of the command changes slightly. Chipset will still send the number of bytes (on writes) or receive the number of bytes (on reads) indicated in the DATA0 register. However, it will not send the contents of the DATA0 register as part of the message. Also, the Block Write protocol sequence changes slightly: the Byte Count (bits 27:20 in the bit sequence) are not sent - as a result, the slave will not acknowledge (bit 28 in the sequence).

### $I^2C$ Read

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

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

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

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

The format that is used for the command is shown in Table 5-76.

**Table 5-76.$I^2C$ Block Read**

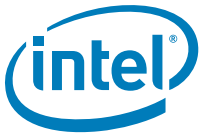
| Bit | Description |
|---|---|
| 1 | Start |
| 8:2 | Slave Address — 7 bits |
| 9 | Write |
| 10 | Acknowledge from slave |
| 18:11 | Send DATA1 register |
| 19 | Acknowledge from slave |
| 20 | Repeated Start |
| 27:21 | Slave Address — 7 bits |
| 28 | Read |
| 29 | Acknowledge from slave |
| 37:30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

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

### Block Write–Block Read Process Call

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

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



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

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

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

Note that there is no STOP condition before the repeated START condition, and that a NACK signifies the end of the read transfer.

***Note:*** E32B bit in the Auxiliary Control register must be set when using this protocol.

See section 5.5.8 of the *System Management Bus (SMBus) Specification,* Version 2.0 for the format of the protocol.

### 5.20.2 Bus Arbitration

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

If Chipset sees that it has lost arbitration, the condition is called a collision. Chipset will set the BUS_ERR bit in the Host Status Register, and if enabled, generate an interrupt or SMI#. The processor is responsible for restarting the transaction.

When Chipset is a SMBus master, it drives the clock. When Chipset is sending address or command as an SMBus master, or data bytes as a master on writes, it drives data relative to the clock it is also driving. It will not start toggling the clock until the start or stop condition meets proper setup and hold time. Chipset will also provide minimum time between SMBus transactions as a master.

***Note:*** Chipset supports the same arbitration protocol for both the SMBus and the System Management (SMLINK) interfaces.

### 5.20.3 Bus Timing

#### 5.20.3.1 Clock Stretching

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

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

#### 5.20.3.2 Bus Time Out (Chipset as SMBus Master)

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

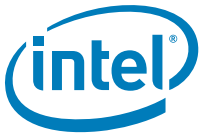
The 25 ms timeout counter will not count under the following conditions:

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).

### 5.20.4 Interrupts / SMI#

Chipset 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 (Device 31:Function 0:Offset 40h:bit 1).

Table 5-78 and Table 5-79 specify how the various enable bits in the SMBus function control the generation of the interrupt, Host and Slave SMI, and Wake internal signals. The rows in the tables are additive, which means that if more than one row is true for a particular scenario then the Results for all of the activated rows will occur.



### Table 5-77.Enable for SMBALERT#

| Event | INTREN (Host Control I/O Register, Offset 02h, Bit 0) | SMB_SMI_EN (Host Configuration Register, D31:F3:Offset 40h, Bit 1) | SMBALERT_DIS (Slave Command I/O Register, Offset 11h, Bit 2) | Result |
|---|---|---|---|---|
| SMBALERT# asserted low (always reported in Host Status Register, Bit 5) | X | X | X | Wake generated |
| | X | 1 | 0 | Slave SMI# generated (SMBUS_SMI_STS) |
| | 1 | 0 | 0 | Interrupt generated |

### Table 5-78.Enables for SMBus Slave Write and SMBus Host Events

| Event | INTREN (Host Control I/O Register, Offset 02h, Bit 0) | SMB_SMI_EN (Host Configuration Register, D31:F3:Offset 40h, Bit1) | Event |
|---|---|---|---|
| Slave Write to Wake/SMI# Command | X | X | Wake generated when asleep.<br>Slave SMI# generated when awake (SMBUS_SMI_STS). |
| Slave Write to SMLINK_SLAVE_SMI Command | X | X | Slave SMI# generated when in the S0 state (SMBUS_SMI_STS) |
| Any combination of Host Status Register [4:1] asserted | 0 | X | None |
| | 1 | 0 | Interrupt generated |
| | 1 | 1 | Host SMI# generated |

### Table 5-79.Enables for the Host Notify Command

| HOST_NOTIFY_INTREN (Slave Control I/O Register, Offset 11h, bit 0) | SMB_SMI_EN (Host Config Register, D31:F3:Off40h, Bit 1) | HOST_NOTIFY_WKEN (Slave Control I/O Register, Offset 11h, bit 1) | Result |
|---|---|---|---|
| 0 | X | 0 | None |
| X | X | 1 | Wake generated |
| 1 | 0 | X | Interrupt generated |
| 1 | 1 | X | Slave SMI# generated (SMBUS_SMI_STS) |

### 5.20.5 SMBALERT#

SMBALERT# is multiplexed with GPIO11. When enable and the signal is asserted, Chipset can generate an interrupt, an SMI#, or a wake event from S1–S5.

***Note:*** Any event on SMBALERT# (regardless whether it is programmed as a GPI or not), causes the event message to be sent in heartbeat mode.

### 5.20.6 SMBus CRC Generation and Checking

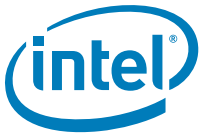
If the AAC bit is set in the Auxiliary Control register, Chipset automatically calculates and drives CRC at the end of the transmitted packet for write cycles, and will check the CRC for read cycles. It will not transmit the contents of the PEC register for CRC. The PEC bit must not be set in the Host 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.

### 5.20.7 SMBus Slave Interface

Chipset's SMBus Slave interface is accessed via the SMBus. The SMBus slave logic will not generate or handle receiving the PEC byte and will only act as a Legacy Alerting Protocol device. The slave interface allows Chipset to decode cycles, and allows an external microcontroller to perform specific actions. Key features and capabilities include:

- Supports decode of three types of messages: Byte Write, Byte Read, and Host Notify.
- Receive Slave Address register: This is the address that Chipset decodes. A default value is provided so that the slave interface can be used without the processor having to program this register.
- Receive Slave Data register in the SMBus I/O space that includes the data written by the external microcontroller.
- Registers that the external microcontroller can read to get the state of Chipset.
- Status bits to indicate that the SMBus slave logic caused an interrupt or SMI# due to the reception of a message that matched the slave address.
  - Bit 0 of the Slave Status Register for the Host Notify command
  - Bit 16 of the SMI Status Register (Section 13.8.3.13) for all others



***Note:*** The external microcontroller should not attempt to access chipset's SMBus slave logic until either:

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

The 800 ms case is based on the scenario where the RTC Battery is dead or missing such that the RTC Power Well comes up simultaneously with Suspend Well. In this case, the RTC clock may take a while to stabilize. Chipset uses the RTC clock to extend the internal RSMRST# by ~100 ms. Therefore, if the clock is slow to toggle, this time could be extended. 800 ms is assumed to be sufficient guardband for this.

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

***Note:*** When an external microcontroller accesses the SMBus Slave Interface over the SMBus a translation in the address is needed to accommodate the least significant bit used for read/write control. For example, if Chipset slave address (RCV_SLVA) is left at 44h (default), the external micro controller would use an address of 88h/89h (write/read).

### 5.20.7.1 Format of Slave Write Cycle

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

***Note:*** If Chipset is sent a 'Hard Reset Without Cycling' command on SMBus while the system is in S4 or S5, the reset command will not be executed until the next wake event. SMBus write commands sent after the Hard Reset Without Cycling command and before the wake event will be NAKed by Chipset. This also applies to any SMBus wake commands sent after a Hard Reset Without Cycling command, such that the SMBus wake command will not cause the system to wake. Any SMBus read that is accepted by Chipset will complete normally.

Table 5-80 has the values associated with the registers.

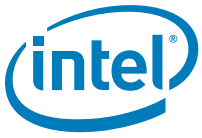
**Table 5-80.Slave Write Registers**

| Register | Function |
|---|---|
| 0 | Command Register. See Table 5-81 below for legal values written to this register. |
| 1–3 | **Reserved** |
| 4 | Data Message Byte 0 |
| 5 | Data Message Byte 1 |
| 6–7 | **Reserved** |
| 8 | **Reserved** |
| 9–FFh | **Reserved** |

**NOTE:** The external microcontroller is responsible to make sure that it does not update the contents of the data byte registers until they have been read by the system processor. Chipset 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. Chipset will not attempt to cover this race condition (i.e., unpredictable results in this case).

### Table 5-81.Command Types

| Command Type | Description |
|---|---|
| 0 | **Reserved** |
| 1 | **WAKE/SMI#.** This command wakes the system if it is not already awake. If system is already awake, an SMI# is generated.<br>**NOTE:** The SMB_WAK_STS bit will be set by this command, even if the system is already awake. The SMI handler should then clear this bit. |
| 2 | **Unconditional Powerdown.** This command sets the PWRBTNOR_STS bit, and has the same effect as the Powerbutton Override occurring. |
| 3 | **HARD RESET WITHOUT CYCLING:** This command causes a hard reset of the system (does not include cycling of the power supply). This is equivalent to a write to the CF9h register with bits 2:1 set to 1, but bit 3 set to 0. |
| 4 | **HARD RESET SYSTEM.** This command causes a hard reset of the system (including cycling of the power supply). This is equivalent to a write to the CF9h register with bits 3:1 set to 1. |
| 5 | **Disable the TCO Messages.** This command will disable Chipset from sending Heartbeat and Event messages (as described in Section 5.15.2). Once this command has been executed, Heartbeat and Event message reporting can only be re-enabled by assertion and deassertion of the RSMRST# signal. |
| 6 | **WD RELOAD:** Reload watchdog timer. |
| 7 | **Reserved** |
| 8 | **SMLINK_SLV_SMI.** When Chipset detects this command type while in the S0 state, it sets the SMLINK_SLV_SMI_STS bit (see Section 13.9.5). This command should only be used if the system is in an S0 state. If the message is received during S1–S5 states, Chipset acknowledges it, but the SMLINK_SLV_SMI_STS bit does not get set.<br>Note: It is possible that the system transitions out of the S0 state at the same time that the SMLINK_SLV_SMI command is received. In this case, the SMLINK_SLV_SMI_STS bit may get set but not serviced before the system goes to sleep. Once the system returns to S0, the SMI associated with this bit would then be generated. Software must be able to handle this scenario. |
| 9–FFh | **Reserved** |



### 5.20.7.2 Format of Read Command

The external master performs Byte Read commands to Chipset SMBus Slave I/F. 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 Table 5-82 shows the Read Cycle Format. Table 5-83 shows the register mapping for the data byte.

**Table 5-82.Read Cycle Format**

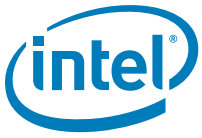
| Bit | Description | Driven by | Comment |
|---|---|---|---|
| 1 | Start | External Microcontroller | |
| 8:2 | Slave Address - 7 bits | External Microcontroller | Must match value in Receive Slave Address register |
| 9 | Write | External Microcontroller | Always 0 |
| 10 | ACK | Chipset | |
| 18:11 | Command code - 8 bits | External Microcontroller | Indicates which register is being accessed See Table 5-83 |
| 19 | ACK | Chipset | |
| 20 | Repeated Start | External Microcontroller | |
| 27:21 | Slave Address - 7 bits | External Microcontroller | Must match value in Receive Slave Address register |
| 28 | Read | External Microcontroller | Always 1 |
| 29 | ACK | Chipset | |
| 37:30 | Data Byte | Chipset | Value depends on register being accessed. See Table 5-83 |
| 38 | NOT ACK | External Microcontroller | |
| 39 | Stop | External Microcontroller | |

**Table 5-83.Data Values for Slave Read Registers**

| Register | Bits | Description |
|---|---|---|
| 0 | 7:0 | **Reserved** |
| 1 | 2:0 | System Power State<br>000 = S0<br>001 = S1<br>010 = **Reserved**<br>011 = S3<br>100 = S4<br>101 = S5<br>110 = **Reserved**<br>111 = **Reserved** |
| 1 | 7:3 | **Reserved** |
| 2 | 3:0 | Frequency Strap Register |
| 2 | 7:4 | **Reserved** |

### Table 5-83.Data Values for Slave Read Registers

| Register | Bits | Description |
|---|---|---|
| 3 | 5:0 | Watchdog Timer current value |
| 3 | 7:6 | **Reserved** |
| 4 | 0 | 1 = The Intruder Detect (INTRD_DET) bit is set. This indicates that the system cover has probably been opened. |
| 4 | 1 | 1 = BTI Temperature Event occurred. This bit will be set if chipset's THRM# input signal is active. Need to take after polarity control. |
| 4 | 2 | Boot-status. This bit will be 1 when the processor does not fetch the first instruction. |
| 4 | 3 | This bit will be set after the TCO timer times out a second time (Both TIMEOUT and SECOND_TO_STS bits set). |
| 4 | 6:4 | **Reserved** |
| 4 | 7 | The bit will reflect the state of the GPI11/SMBALERT# signal, and will depend on the GP_INV11 bit. It does not matter if the pin is configured as GPI11 or SMBALERT#.<br>• If the GP_INV11 bit is 1, the value of register 4 bit 7 will equal the level of the GPI11/SMBALERT# pin (high = 1, low = 0).<br>• If the GP_INV11 bit is 0, the value of register 4 bit 7 will equal the inverse of the level of the GPI11/SMBALERT# pin (high = 1, low = 0). |
| 5 | 0 | Unprogrammed flash BIOS bit. This bit will be 1 to indicate that the first BIOS fetch returned FFh, that indicates that the flash BIOS is probably blank. |
| 5 | 1 | Battery Low Status (Netbook Only). 1 if BATLOW# is '0' |
| 5 | 2 | Processor Power Failure Status. 1 if the CPUPWR_FLR bit in the GEN_PMCON_2 register is set. |
| 5 | 3 | INIT# due to receiving Shutdown message. This event is visible from the reception of the shutdown message until a platform reset is done. Events on signal will not create an event message. |
| 5 | 4 | LT Range: LT reset indication. Events on signal will not create an event message. |
| 5 | 5 | POWER_OK_BAD: Indicates the failure core power well ramp during boot/resume. This bit will be active if the SLP_S3# pin is de-asserted and PWROK pin is not asserted. |
| 5 | 6 | Thermal Trip: This bit will shadow the state of CPU Thermal Trip status bit (CTS). Events on signal will not create an event message. |
| 5 | 7 | **Reserved** |
| 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. |



#### 5.20.7.2.1 Behavioral Notes

According to SMBus protocol, Read and Write messages always begin with a Start bit – Address– Write bit sequence. When Chipset detects that the address matches the value in the Receive Slave Address register, it will assume that the protocol is always followed and ignore the Write bit (bit 9) and signal an Acknowledge during bit 10. In other words, if a Start –Address–Read occurs (which is invalid for SMBus Read or Write protocol), and the address matches chipset's Slave Address, Chipset 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 chipset's Receive Slave Address, it will assume that the protocol is followed, ignore bit 28, and proceed with the Slave Read cycle.

***Note:*** An external microcontroller must not attempt to access chipset's SMBus Slave logic until at least 1 second after both RTCRST# and RSMRST# are deasserted (high).

### 5.20.7.3 Format of Host Notify Command

Chipset 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 Chipset already has data for a previously-received host notify command which has not been serviced yet by the host software (as indicated by the HOST_NOTIFY_STS bit), then it will NACK following the host address byte of the protocol. This allows the host to communicate non-acceptance to the master and retain the host notify address and data values for the previous cycle until host software completely services the interrupt.

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

Table 5-84 shows the Host Notify format.

**Table 5-84.Host Notify Format (Sheet 1 of 2)**

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

**Table 5-84.Host Notify Format (Sheet 2 of 2)**

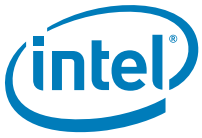
| Bit | Description | Driven By | Comment |
|---|---|---|---|
| 19 | ACK | Chipset | |
| 27:20 | Data Byte Low — 8 bits | External Master | Loaded into the Notify Data Low Byte Register |
| 28 | ACK | Chipset | |
| 36:29 | Data Byte High — 8 bits | External Master | Loaded into the Notify Data High Byte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

## 5.21 Intel HD Audio Overview

chipset's controller communicates with the external codec(s) over the Intel High Definition Audio serial link. The controller consists of a set of DMA engines that are used to move samples of digitally encoded data between system memory and an external codec(s). Chipset implements four output DMA engines and 4 input DMA engines. The output DMA engines move digital data from system memory to a D-A converter in a codec. Chipset implements a single Serial Data Output signal (HDA_SDOUT) that is connected to all external codecs. The input DMA engines move digital data from the A-D converter in the codec to system memory. Chipset implements three Serial Digital Input signals (HDA_SDI[2:0]) supporting up to three codecs.

Audio software renders outbound and processes inbound data to/from buffers in system memory. The location of individual buffers is described by a Buffer Descriptor List (BDL) that is fetched and processed by the controller. The data in the buffers is arranged in a predefined format. The output DMA engines fetch the digital data from memory and reformat it based on the programmed sample rate, bit/sample and number of channels. The data from the output DMA engines is then combined and serially sent to the external codecs over the Intel High Definition Audio link. The input DMA engines receive data from the codecs over the Intel High Definition Audio link and format the data based on the programmable attributes for that stream. The data is then written to memory in the predefined format for software to process. Each DMA engine moves one stream of data. A single codec can accept or generate multiple streams of data, one for each A-D or D-A converter in the codec. Multiple codecs can accept the same output stream processed by a single DMA engine.

Codec commands and responses are also transported to and from the codecs via DMA engines.



## 5.22 Serial Peripheral Interface (SPI)

The Serial Peripheral Interface (SPI) is a 4-pin interface that provides a potentially lower-cost alternative for system flash versus the Firmware Hub on the LPC bus.

The 4-pin SPI interface consists of clock (CLK), master data out (Master Out Slave In (MOSI)), master data in (Master In Slave Out (MISO)) and an active low chip select (CS#).

Communication on the SPI bus is done with a Master – Slave protocol. The typical bus topology consists of a single SPI Master with a single SPI Slave (flash device). The Slave is connected to Chipset and is implemented as a tri-state bus.

***Note:*** When SPI is selected by the Boot BIOS Destination Strap and a SPI device is detected by Chipset, LPC based BIOS flash is disabled.

### 5.22.1 Flash Device Configurations

Chipset SPI flash may be used in two configurations. Table 5-85 focuses on these various configurations involving Chipset.

**Table 5-85.SPI Implementation Options**

| Configuration | System BIOS Location | System BIOS and Intel AMT Shared Flash | FWH Present | Number of SPI Device(s) |
|---|---|---|---|---|
| 1 | FWH | No | Yes | 0 |
| 2 | SPI | No | No | 1 |

***Note:*** Chipset SPI interface supports a single Chip Select pin for a single SPI device.

### 5.22.2 SPI Device Compatibility Requirements

A variety of SPI flash devices exist in the market. In order for a SPI device to be compatible with Chipset it must meet the minimum requirements detailed in the following sections.

#### 5.22.2.1 Chipset SPI Based BIOS Only Configuration Requirements

A SPI flash device must meet the following minimum requirements to be compatible with Chipset in a non-shared flash configuration:

- Erase size capability of at least one of the following: 64 KB, 32 KB, 4 KB, 2 KB, 512 bytes, or 256 bytes.
- Required command set and associated opcodes (Refer to Section 5.22.3.1).
- Device identification command (Refer to Section 5.22.3.2).
- Device must support multiple writes to a page without requiring a preceding erase cycle (Refer to Section 5.22.4)

- Serial flash device must ignore the upper address bits such that an address of FFFFFFh simply aliases to the top of the flash memory.
- SPI Compatible Mode 0 support (clock phase is 0 and data is latched on the rising edge of the clock).
- If the device receives a command that is not supported, the device must complete the cycle gracefully without any impact on the flash content.
- An erase command (page, sector, block, chip, or etc.) must set to 1 (FFh) all bits inside the designated area (page, sector, block, chip, or etc.).
- Minimum density of 4 Mb (Platform dependent based on size of BIOS).

***Note:*** Chipset only supports Mode 0 on SPI flash devices

## 5.22.3 Chipset Compatible Command Set

### 5.22.3.1 Required Command Set for Inter Operability

Table 5-86 contains a list of commands and the associated opcodes that a SPI based serial flash device must support in order to be interoperable with the serial flash interface.

**Table 5-86.Required Commands and Opcodes**

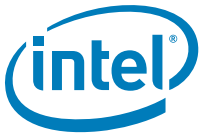
| Commands | OPCODE |
|---|---|
| Program Data | 02h |
| Read Data | 03h |
| Read Status | 05h |

### 5.22.3.2 Recommended Standard Commands

The following table contains a list of standard commands that a SPI device should support to be compatible with Chipset. This list only contains standard commands and is not meant to be an all inclusive list of commands that SPI devices can support.

**Table 5-87.Chipset Standard SPI Commands**

| Commands | OPCODE | Notes |
|---|---|---|
| Write Status | 01h | If command is supported by a device, 01h must be supported. |
| Write Disable | 04h | |
| Write Enable | 06h | If command is supported by a device, 06h must be supported. |
| Fast Read | 0Bh | Chipset does not support this command. |
| JEDEC ID | 9Fh | Either JEDEC ID (9Fh) or an Identify Device with ABh is required, not both. |
| Identify Device | ABh | Either JEDEC ID (9Fh) or an Identify Device with ABh is required, not both |



### 5.22.4 Flash Protection

There are three types of Flash Protection mechanisms:

1. BIOS Range Write Protection
2. SMI#-Based Global Write Protection

The three mechanisms are conceptually OR'd together such that if any of the mechanisms indicate that the access should be blocked, then it is blocked. Table 5-88 provides a summary of the Three Mechanisms.

**Table 5-88.Flash Protection Mechanism Summary**

| Mechanism | Accesses Blocked | Range Specific ? | Reset-Override or SMI#-Override? | Equivalent Function on FWH |
|---|---|---|---|---|
| BIOS Range Write Protection | Writes | Yes | Reset Override | FWH Sector Protection |
| Write Protect | Writes | No | SMI# Override | Same as Write Protect in previous ICH components for FWH |
| BIOS BAR | Reads and Writes | Yes | Reset Override | Not Applicable- Specific to Flash Sharing |

A blocked command will appear to software to finish, except that the Blocked Access status bit is set in this case.

#### 5.22.4.1 BIOS Range Write Protection

Chipset provides a method for blocking writes to specific ranges in the SPI flash when the Protected BIOS Ranges are enabled. This is achieved by checking the Opcode type information (which can be locked down by the initial Boot BIOS) and the address of the requested command against the base and limit fields of a Write Protected BIOS range.

***Note:*** Once BIOS has locked down the Protected BIOS Range registers, this mechanism remains in place until the next system reset.

#### 5.22.4.2 SMI# Based Global Write Protection

Chipset provides a method for blocking writes to the SPI flash when the Write Protect bit is cleared (i.e., protected). This is achieved by checking the Opcode type information (which can be locked down by the initial Boot BIOS) of the requested command.

The Write Protect and Lock Enable bits interact in the same manner for SPI BIOS as they do for the FWH BIOS.

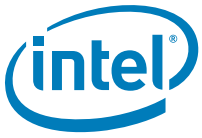
## 5.23 Feature Capability Mechanism

A new set of registers have been added into Chipset LPC Interface (Device 31, Function 0, offset E0h - EBh) that allows the system software or BIOS to easily determine the features supported by Chipset. These registers can be accessed through LPC PCI configuration space; thus allowing for convenient single point access mechanism for chipset feature detection.

This set of registers consists of:

- Capability ID (FDCAP)
- Capability Length (FDLEN)
- Capability Version and Vendor-Specific Capability ID (FDVER)
- Feature Vector (FVECT)

§



# 6 Ballout Definition

## 6.1 Chipset Ballout, Signal, and Mechanical Document

An electronic version of the ballout is available. Please look for Chipset *Ballout, Signal, and Mechanical Document* on IBL with Doc ID: 405447

## 6.2 Chipset Ballout

This section contains Chipset ballout information. The ballout is preliminary and subject to change. The figures below are the ballout map of the 360 BGA package. Figure 6-16 is a ballout list, sorted alphabetically by signal name.

## Figure 6-16. Chipset Ballout (Topview–Left Side)

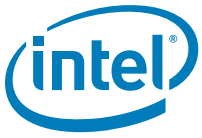
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | VSS | STRAP2 # / | AD22 | --- | PAR | AD24 | --- | PLOCK# | --- | TRDY# | --- | --- | AD10 |
| B | AD31 | PIRQA# | PIRQC# | --- | AD23 | VSS | IRDY# | AD21 | AD18 | VSS | SERR# | --- | AD19 |
| C | AD30 | OC6# / GPIO30 | OC7# / GPIO31 | --- | OC1# | --- | AD29 | AD27 | GPIO1 | --- | AD16 | --- | C/BE2# |
| D | --- | OC3# | OC2# | OC0# | --- | PIRQF# / GPIO3 | PIRQB# | --- | AD28 | PERR# | STRAP0 # | --- | |
| E | --- | --- | --- | --- | OC4# | OC5# / GPIO29 | --- | PIRQE# / GPIO2 | --- | AD15 | --- | AD17 | --- |
| F | VccSus 3_3 | Vss | --- | CLK48 | V5REF_ SUS | VCCUS BPLL | --- | PIRQH# / GPIO5 | --- | Vcc3_3 | --- | V5REF | --- |
| G | --- | USBRBI AS | USBRBI AS# | VSS | --- | --- | --- | VSS | --- | Vcc3_4 | --- | AD25 | --- |
| H | VSS | USBP1P | USBP1N | VSS | VSS | USBP0P | USBP0N | PIRQG# / GPIO4 | --- | PIRQD# | --- | AD26 | --- |
| J | --- | USBP2N | USBP2P | --- | --- | --- | --- | --- | --- | Vcc1_0 5 | --- | PCICLK | --- |
| K | USBP4N | USBP4P | --- | VSS | USBP3P | USBP3N | VccSus 3_3 | VSS | RSVD | --- | VSS | --- | --- |
| L | --- | USBP5N | USBP5P | VSS | --- | --- | --- | --- | --- | --- | --- | AD20 | --- |
| M | --- | --- | --- | --- | USBP6P | USBP6N | VSS | SPI_CS# | VCC1_5 | --- | VSS | --- | RSVD |
| N | USBP7N | USBP7P | VSS | VccSus 3_3 | --- | --- | --- | --- | --- | --- | --- | VSS | VSS |
| P | --- | --- | --- | --- | GPIO14 | HDA_BI T_CLK | LAN_RS TSYNC | HDA_SD IN2 | SPI_CLK | --- | VSS | --- | VSS |
| R | --- | SPI_MIS O | GPIO24 | SPI_AR B | --- | --- | --- | --- | --- | Vcc3_3 | --- | RSVD | --- |
| T | SPI_MO SI | VSS | --- | LAN_CL K | RTCRST # | EE_DOU T | LAN_TX D1 | INTRUDE R# | Vcc3_3 | --- | --- | --- | --- |
| U | --- | HDA_RS T# | EE_CS | LAN_TX D2 | --- | --- | --- | --- | --- | PWROK | --- | RSVD | --- |
| V | VSS | HDA_SD IN1 | EE_SHC LK | --- | RTCX2 | LAD0 / FWH0 | VSS | VSS | --- | Vcc1_0 5 | --- | RSVD | --- |
| W | --- | HDA_SD IN0 | LAN_TX D0 | RTCX1 | --- | --- | --- | LAD3 / FWH3 | --- | RSVD | --- | VSS | --- |
| Y | HDA_S YNC | VSS | --- | FWH4 /LFRAM | LAD2 / FWH2 | VCCAPL L | --- | LDRQ0# | --- | RSVD | --- | RSVD | --- |
| AA | HDA_SD OUT | LAN_RX D0 | CLK14 | --- | LDRQ1# / GPIO23 | LAD1 / FWH1 | --- | VCC1_5 | --- | RSVD | --- | RSVD | --- |
| AB | --- | --- | --- | VSS | --- | VSS | VSS | VSS | --- | RSVD | RSVD | --- | RSVD |
| AC | GPIO34 | LAN_RX D2 | RSMRST # | SATA_C LKP | --- | --- | SATA0T XN | VSS | SATA1T XP | --- | SATAR BIAS | --- | RSVD |
| AD | LAN_RX D1 | VSS | INTVRM EN | SATA_C LKN | --- | SATA0R XP | SATA0T XP | SATA1R XP | SATA1T XN | VSS | SATAR BIAS# | --- | Vcc3_3 |
| AE | VSS | EE_DIN | VCCRTC | --- | --- | SATA0R XN | --- | SATA1R XN | --- | VSS | --- | --- | VSS |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

## Figure 6-17.Chipset Ballout (Topview–Right Side)

| 10 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TRDY# | AD10 | --- | --- | FRAME# | --- | GNT1# | --- | REQ2# | --- | --- | PCIRST# | GPIO12 | VSS | A |
| VSS | AD19 | --- | DEVSEL# | VSS | AD4 | AD6 | AD7 | VSS | --- | AD0 | LAN_RST# | VSS | BATLOW# | B |
| --- | C/BE2# | --- | GPIO22 | --- | AD2 | AD3 | AD5 | --- | --- | PME# | GPIO13 | GPIO25 | WAKE# | C |
| PERR# | | --- | AD9 | AD8 | --- | AD1 | GPIO26 | GPIO27 | --- | SUSCLK | --- | --- | --- | D |
| AD15 | --- | AD11 | --- | GNT2# | --- | VSS | --- | SMBALERT# / | PWRBTN# | --- | SMBDATA | GPIO15 | SLP_S4# | E |
| Vcc3_3 | --- | STOP# | --- | VSS | --- | VccSus3_3 | --- | TP3 | SLP_S5# | GPIO28 | --- | SMLINK1 | SMLINK0 | F |
| Vcc3_4 | --- | STRAP1# / | --- | REQ1# | --- | SYS_RESET# | --- | --- | --- | SUS_STAT# / | PLTRST# | Vss | --- | G |
| PIRQD# | --- | AD12 | --- | C/BE0# | --- | SMBCLK | GPIO9 | SLP_S3# | LINKALERT# | --- | RI# | DMI_ZCOMP | Vcc3_3 | H |
| Vcc1_05 | --- | AD14 | --- | SPKR | --- | --- | --- | --- | --- | DMI_IRCOMP | PETn1 | PETp1 | --- | J |
| --- | --- | --- | --- | --- | Vcc1_05 | GPIO8 | VSS | VSS | PERn1 | PERp1 | --- | PERn2 | PERp2 | K |
| --- | --- | AD13 | --- | C/BE3# | --- | --- | --- | --- | --- | PETn3 | PERn3 | PERp4 | --- | L |
| --- | RSVD | --- | C/BE1# | --- | GPIO10 | PERn2 | PERp2 | VCC1_5 | PETp3 | --- | --- | --- | --- | M |
| --- | VSS | VSS | --- | --- | --- | --- | --- | --- | --- | VCC1_5 | VSS | PETp4 | PETn4 | N |
| --- | VSS | --- | Vcc1_05 | --- | PERn4 | PERp4 | VSS | DMI0TXP | DMI0TXN | --- | --- | --- | --- | P |
| Vcc3_3 | --- | VSS | --- | --- | --- | --- | --- | --- | --- | VSS | DMI0RXN | DMI0RXP | --- | R |
| --- | --- | --- | BMBUSY# / | --- | NMI | DMI2RXP | DMI2RXN | DMI1RXP | DMI1RXN | VSS | --- | DMI1TXN | DMI1TXP | T |
| PWROK | --- | GPIO33 | --- | A20GATE | --- | --- | --- | --- | --- | --- | DMI2TXN | DMI2TXP | --- | U |
| Vcc1_05 | --- | RSVD | --- | VRMPWRGD | --- | STPCLK# | VSS | DMI3RXP | DMI3RXN | VSS | DMI3TXP | DMI3TXN | VSS | V |
| RSVD | --- | GPIO7 | --- | GPIO6 | --- | V_CPU_IO | --- | --- | --- | VSS | DMI_CLKN | DMI_CLKP | --- | W |
| RSVD | --- | RSVD | --- | STP_PCI# | --- | IGNNE# | --- | A20M# | CPUSLP# | FERR# | --- | VSS | VccDMIPLL | Y |
| RSVD | --- | RSVD | --- | SERIRQ | --- | DPSLP# | --- | THRMTRIP# | SMI# | --- | --- | --- | --- | AA |
| RSVD | RSVD | --- | RSVD | RSVD | THRM# | --- | STP_CPU# | DPRSLPVR / | --- | CPUPWRGD / | DPRSTP# | INTR | --- | AB |
| --- | RSVD | --- | RSVD | --- | RSVD | MCH_SYNC# | CLKRUN# | --- | RCIN# | --- | GPIO38 | GPIO39 | INIT# | AC |
| VSS | Vcc3_3 | --- | RSVD | RSVD | RSVD | RSVD | RSVD | VSS | INIT3_3V# | --- | GPIO36 | VSS | SATALED# | AD |
| VSS | VSS | --- | --- | RSVD | --- | RSVD | --- | RSVD | RSVD | --- | RSVD | RSVD | VSS | AE |
| 10 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | |

### Table 6-89.Chipset Ballout by Signal Name (Sheet 1 of 4)

| Signal Name | Ball # | Signal Name | Ball # | Signal Name | Ball # |
|---|---|---|---|---|---|
| A20GATE | U16 | AD27 | C8 | RSVD | AD15 |
| A20M# | Y20 | AD28 | D9 | RSVD | AC17 |
| HDA_BIT_CLK | P6 | AD29 | C7 | RSVD | AB13 |
| HDA_RST# | U2 | AD30 | C1 | DEVSEL# | B15 |
| HDA_SDIN0 | W2 | AD31 | B1 | TP3 | F20 |
| HDA_SDIN1 | V2 | LAN_RST# | B23 | RSVD | AC13 |
| HDA_SDIN2 | P8 | BATLOW# | B25 | RSVD | AB15 |
| HDA_SDOUT | AA1 | CLK14 | AA3 | DMI_CLKN | W23 |
| HDA_SYNC | Y1 | CLK48 | F4 | DMI_CLKP | W24 |
| AD0 | B22 | CLKRUN# | AC19 | DMI_ZCOMP | H24 |
| AD1 | D18 | CPUPWRGD / GPIO49 | AB22 | DMI_IRCOMP | J22 |
| AD2 | C17 | CPUSLP# | Y21 | DMI0RXN | R23 |
| AD3 | C18 | C/BE0# | H16 | DMI0RXP | R24 |
| AD4 | B17 | C/BE1# | M15 | DMI0TXN | P21 |
| AD5 | C19 | C/BE2# | C13 | DMI0TXP | P20 |
| AD6 | B18 | C/BE3# | L16 | DMI1RXN | T21 |
| AD7 | B19 | RSVD | AB16 | DMI1RXP | T20 |
| AD8 | D16 | RSVD | AE24 | DMI1TXN | T24 |
| AD9 | D15 | RSVD | AE23 | DMI1TXP | T25 |
| AD10 | A13 | RSVD | AA14 | DMI2RXN | T19 |
| AD11 | E14 | RSVD | V14 | DMI2RXP | T18 |
| AD12 | H14 | RSVD | R12 | DMI2TXN | U23 |
| AD13 | L14 | RSVD | AE20 | DMI2TXP | U24 |
| AD14 | J14 | RSVD | W10 | DMI3RXN | V21 |
| AD15 | E10 | RSVD | V12 | DMI3RXP | V20 |
| AD16 | C11 | RSVD | AE21 | DMI3TXN | V24 |
| AD17 | E12 | RSVD | AE18 | DMI3TXP | V23 |
| AD18 | B9 | RSVD | AD19 | DPRSLPVR | AB20 |
| AD19 | B13 | RSVD | U12 | DPRSTP# | AB23 |
| AD20 | L12 | RSVD | AD17 | DPSLP# | AA18 |
| AD21 | B8 | RSVD | AC15 | EE_CS | U3 |
| AD22 | A3 | RSVD | AD18 | EE_DIN | AE2 |
| AD23 | B5 | RSVD | Y12 | EE_DOUT | T6 |
| AD24 | A6 | RSVD | AA10 | EE_SHCLK | V3 |
| AD25 | G12 | RSVD | AA12 | FERR# | Y22 |
| AD26 | H12 | RSVD | Y10 | FRAME# | A16 |



### Table 6-89.Chipset Ballout by Signal Name (Sheet 2 of 4)

| Signal Name | Ball # | Signal Name | Ball # | Signal Name | Ball # |
|---|---|---|---|---|---|
| GNT1# | A18 | INTVRMEN | AD3 | PERn2 | M18 |
| GNT2# | E16 | RSVD | Y14 | PERp2 | M19 |
| STRAP0# | D11 | IRDY# | B7 | PETn2 | K24 |
| GPIO48 / STRAP1# | G14 | LAD0 / FWH0 | V6 | PETp2 | K25 |
| BMBUSY# / GPIO0 | T15 | LAD1 / FWH1 | AA6 | PERn3 | L23 |
| GPIO1 | C9 | LAD2 / FWH2 | Y5 | PERp3 | L24 |
| GPIO10 | M17 | LAD3 / FWH3 | W8 | PETn3 | L22 |
| GPIO12 | A24 | LAN_CLK | T4 | PETp3 | M21 |
| GPIO13 | C23 | LAN_RSTSYNC | P7 | PERn4 | P17 |
| GPIO14 | P5 | LAN_RXD0 | AA2 | PERp4 | P18 |
| GPIO15 | E24 | LAN_RXD1 | AD1 | PETn4 | N25 |
| STRAP2# / GPIO17 | A2 | LAN_RXD2 | AC2 | PETp4 | N24 |
| PIRQE# / GPIO2 | E8 | LAN_TXD0 | W3 | PERR# | D10 |
| GPIO24 | R3 | LAN_TXD1 | T7 | PIRQA# | B2 |
| GPIO25 | C24 | LAN_TXD2 | U4 | PIRQB# | D7 |
| GPIO26 | D19 | LDRQ0# | Y8 | PIRQC# | B3 |
| GPIO27 | D20 | LDRQ1# / GPIO23 | AA5 | PIRQD# | H10 |
| GPIO28 | F22 | FWH4/LFRAME# | Y4 | PLOCK# | A8 |
| PIRQF# / GPIO3 | D6 | MCH_SYNC# | AC18 | PLTRST# | G23 |
| GPIO33 | U14 | RSVD | AE16 | PME# | C22 |
| GPIO34 | AC1 | NMI | T17 | PWRBTN# | E21 |
| GPIO36 | AD23 | OC0# | D4 | PWROK | U10 |
| GPIO38 | AC23 | OC1# | C5 | RCIN# | AC21 |
| GPIO39 | AC24 | OC2# | D3 | RSVD | K9 |
| PIRQG# / GPIO4 | H8 | OC3# | D2 | REQ1# | G16 |
| PIRQH# / GPIO5 | F8 | OC4# | E5 | REQ2# | A20 |
| GPIO6 | W16 | OC[5]# / GPIO29 | E6 | RSVD | M13 |
| GPIO7 | W14 | OC[6]# / GPIO30 | C2 | GPIO22 | C15 |
| GPIO8 | K18 | OC[7]# / GPIO31 | C3 | RI# | H23 |
| GPIO9 | H19 | PAR | A5 | RSMRST# | AC3 |
| RSVD | AD16 | PCICLK | J12 | RTCRST# | T5 |
| IGNNE# | Y18 | PCIRST# | A23 | RTCX1 | W4 |
| INIT# | AC25 | PERn1 | K21 | RTCX2 | V5 |
| INIT3_3V# | AD21 | PERp1 | K22 | RSVD | AB10 |
| INTR | AB24 | PETn1 | J23 | SATA_CLKN | AD4 |
| INTRUDER# | T8 | PETp1 | J24 | SATA_CLKP | AC4 |

### Table 6-89.Chipset Ballout by Signal Name (Sheet 3 of 4)

| Signal Name | Ball # | Signal Name | Ball # | Signal Name | Ball # |
|---|---|---|---|---|---|
| SATA0RXN | AE6 | SUSCLK | D22 | V_CPU_IO | W18 |
| SATA0RXP | AD6 | SYS_RESET# | G18 | Vcc3_3 | T9 |
| SATA0TXN | AC7 | THRMTRIP# | AA20 | Vcc3_3 | R10 |
| SATA0TXP | AD7 | THRM# | AB17 | Vcc3_3 | G10 |
| SATA1RXN | AE8 | TRDY# | A10 | Vcc3_3 | F10 |
| SATA1RXP | AD8 | USBP0N | H7 | VccRTC | AE3 |
| SATA1TXN | AD9 | USBP0P | H6 | VccSus3_3 | N4 |
| SATA1TXP | AC9 | USBP1N | H3 | VccSus3_3 | F18 |
| RSVD | AB11 | USBP1P | H2 | VccSus3_3 | K7 |
| SATALED# | AD25 | USBP2N | J2 | Vcc1_5 | AA8 |
| SATARBIAS# | AD11 | USBP2P | J3 | Vcc1_5 | M9 |
| SATARBIAS | AC11 | USBP3N | K6 | VRMPWRGD | V16 |
| SERIRQ | AA16 | USBP3P | K5 | Vss | AD2 |
| SERR# | B11 | USBP4N | K1 | Vss | AE1 |
| SLP_S3# | H20 | USBP4P | K2 | Vss | Y2 |
| SLP_S4# | E25 | USBP5N | L2 | Vss | T2 |
| SLP_S5# | F21 | USBP5P | L3 | Vss | V1 |
| SMBALERT# / GPIO11 | E20 | USBP6N | M6 | Vss | H1 |
| SMBCLK | H18 | USBP6P | M5 | Vss | A1 |
| SMBDATA | E23 | USBP7N | N1 | Vss | AB4 |
| SMI# | AA21 | USBP7P | N2 | Vss | AB6 |
| SMLINK0 | F25 | USBRBIAS# | G3 | Vss | AB7 |
| SMLINK1 | F24 | USBRBIAS | G2 | Vss | L4 |
| LINKALERT# | H21 | Vcc1_05 | V10 | Vss | N3 |
| SPI_ARB | R4 | Vcc1_05 | J10 | Vss | H4 |
| SPI_CLK | P9 | Vcc1_05 | P15 | Vss | H5 |
| SPI_CS# | M8 | Vcc1_05 | K17 | Vss | K4 |
| SPI_MISO | R2 | V5REF | F12 | Vss | G4 |
| SPI_MOSI | T1 | V5REF_Sus | F5 | Vss | B6 |
| SPKR | J16 | Vcc3_3 | H25 | Vss | AD10 |
| STOP# | F14 | Vcc1_5 | M20 | Vss | AE10 |
| STPCLK# | V18 | Vcc1_5 | N22 | Vss | AB8 |
| STP_CPU# | AB19 | VccDMIPLL | Y25 | Vss | AC8 |
| STP_PCI# | Y16 | VccSATAPLL | Y6 | Vss | V7 |
| SUS_STAT# / LPCPD# | G22 | Vcc3_3 | AD13 | Vss | V8 |
| VxxUSBPLL | F6 | VccSus3_3 | F1 | Vss | M7 |

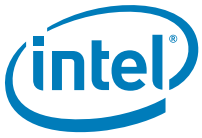


**Table 6-89.Chipset Ballout by Signal Name (Sheet 4 of 4)**

| Signal Name | Ball # | Signal Name | Ball # | Signal Name | Ball # |
|---|---|---|---|---|---|
| Vss | M11 | | | | |
| Vss | P11 | | | | |
| Vss | G8 | | | | |
| Vss | K8 | | | | |
| Vss | K11 | | | | |
| Vss | B10 | | | | |
| Vss | R14 | | | | |
| Vss | W12 | | | | |
| Vss | N12 | | | | |
| Vss | N13 | | | | |
| Vss | N14 | | | | |
| Vss | P13 | | | | |
| Vss | V19 | | | | |
| Vss | P19 | | | | |
| Vss | K19 | | | | |
| Vss | E18 | | | | |
| Vss | F16 | | | | |
| Vss | B16 | | | | |
| Vss | AD20 | | | | |
| Vss | W22 | | | | |
| Vss | T22 | | | | |
| Vss | V22 | | | | |
| Vss | R22 | | | | |
| Vss | K20 | | | | |
| Vss | B20 | | | | |
| Vss | AD24 | | | | |
| Vss | AE25 | | | | |
| Vss | Y24 | | | | |
| Vss | V25 | | | | |
| Vss | N23 | | | | |
| Vss | A25 | | | | |
| Vss | B24 | | | | |
| Vss | G24 | | | | |
| Vss | AE13 | | | | |
| Vss | F2 | | | | |
| WAKE# | C25 | | | | |

§

# 7 Chipset Package Information

The chipset package information is shown in Figure 7-18, Figure 7-19, and Figure 7-20.

***Note:*** All dimensions unless otherwise specified are in millimeters.

**Figure 7-18. Chipset Package (Top View)**

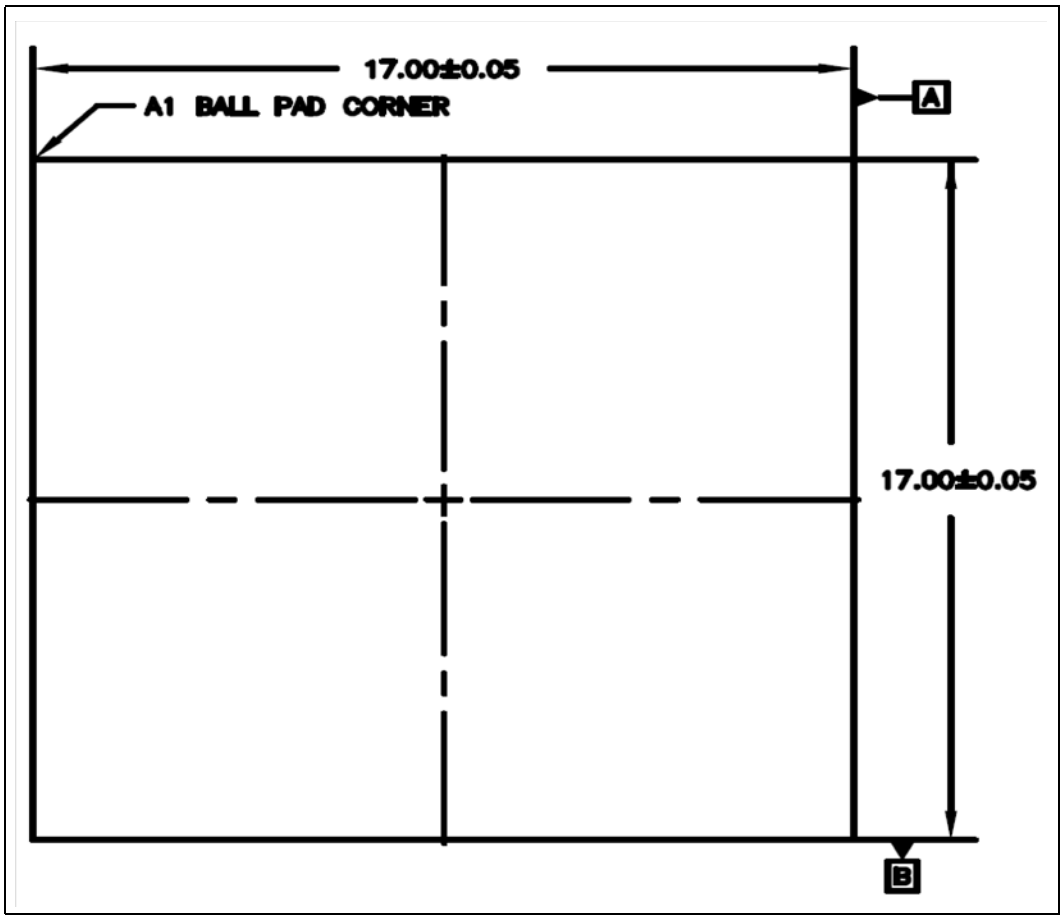


**Figure 7-19. Chipset Package (Bottom View)**

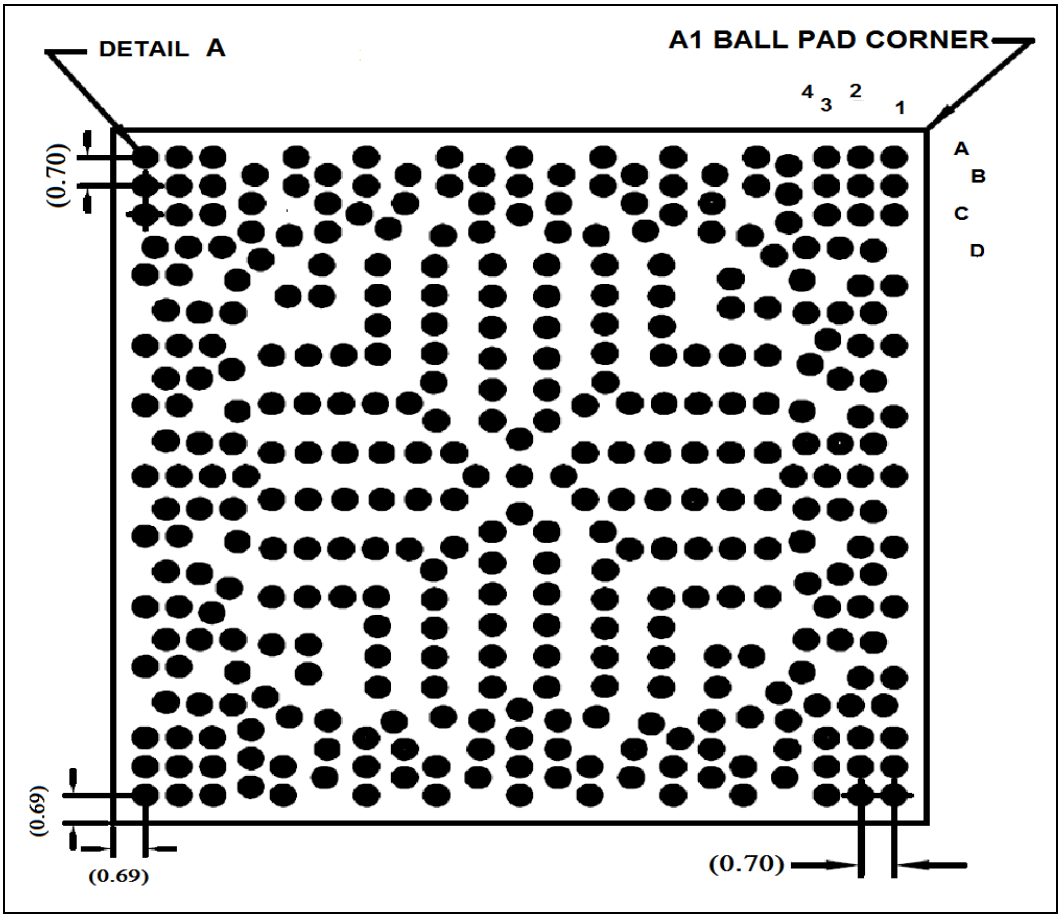
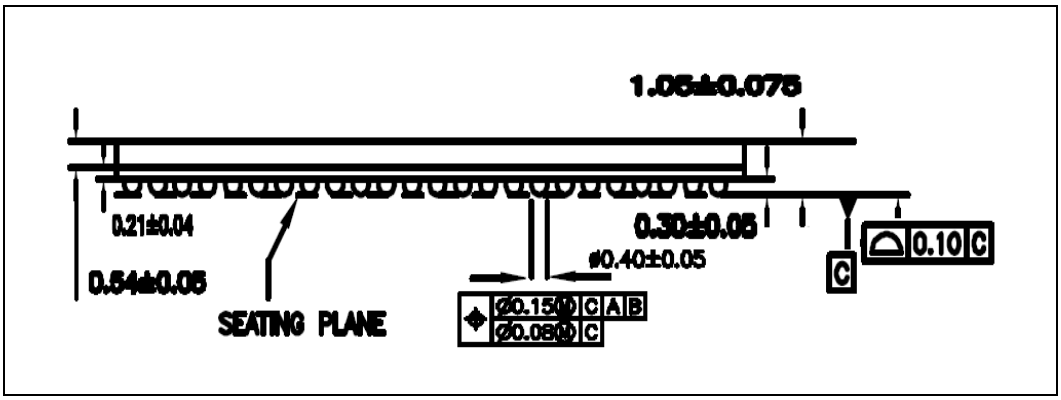



**Figure 7-20. Chipset Package (Side View)**




§

# 8 Electrical Characteristics

This chapter contains the DC and AC characteristics for Chipset. AC timing diagrams are included.

## 8.1 Thermal Specifications

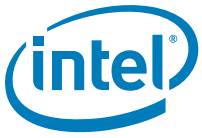
Chipset's Thermal Design Power (TDP) for Nettop platform is 2.1 W and Netbook is 1.5 W.

Refer to chipset Thermal Design Guidelines (Doc ID: 417912) for detail Chipset thermal information.

## 8.2 Absolute Maximum Ratings

**Table 8-90. Chipset Absolute Maximum Ratings**

| Parameter | Maximum Limits |
|---|---|
| Voltage on any 3.3 V Pin with respect to Ground | -0.5 to Vcc3_3 + 0.5 V |
| Voltage on any 5 V Tolerant Pin with respect to Ground (V5REF=5 V) | -0.5 to V5REF + 0.5 V |
| 1.05 V Supply Voltage with respect to VSS | -0.5 to 2.1 V |
| 1.5 V Supply Voltage with respect to VSS | -0.5 to 2.1 V |
| 3.3 V Supply Voltage with respect to VSS | -0.5 to 4.6 V |
| 5.0 V Supply Voltage with respect to VSS | -0.5 to 5.5 V |
| V_CPU_IO Supply Voltage with respect to VSS | -0.5 to 2.1 V |



## 8.3 DC Characteristics

### Table 8-91.DC Current Characteristics

| Power Plane | Maximum Current Consumption | | | |
|---|---|---|---|---|
| **Symbol** | **S0** | **$S3_{COLD}$** | **S4/S5** | **G3** |
| Vcc1_05 | 0.955A | 0.000A | 0.000A | 0.000A |
| Vcc1_5 | 1.422A | 0.000A | 0.000A | 0.000A |
| Vcc3_3[1] | 0.216A | 0.000A | 0.000A | 0.000A |
| VccSus3_3[2] | 0.092A | 27mA | 27mA | 0.000A |
| VccRTC[3] | NA | NA | NA | 6uA |
| V5REF | 6mA | 0.000A | 0.000A | 0.000A |
| V5REF_Sus | 10mA | 10mA | 10mA | 0.000A |
| V_CPU_IO | 14mA | 0.000A | 0.000A | 0.000A |
| VccUSBPLL | 10mA | 0.000A | 0.000A | 0.000A |
| VccDMIPLL | 24mA | 0.000A | 0.000A | 0.000A |
| VccSATAPLL | 45mA | 0.000A | 0.000A | 0.000A |

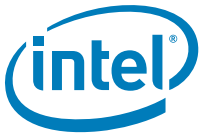
**NOTES:**
1. Intel High Definition Audio codec only supported at 3.3V.
2. VccSus3_3 assumes that the 8 high-speed USB 2.0 devices are connected to chipset's root ports.
3. Icc (RTC) data is estimated with VccRTC at 3.0 V while the system is in a mechanical off (G3) state at room temperature. Only the G3 state of this rail is shown to provide an estimate of battery life.

### Table 8-92.DC Characteristic Input Signal Association (Sheet 1 of 2)

| Symbol | Associated Signals |
|---|---|
| $V_{IH1}/V_{IL1}$ (5V Tolerant) | PCI Signals: AD[31:0], C/BE[3:0]#, DEVSEL#, FRAME#, IRDY#, PAR, PERR#, PLOCK#, REQ[2:1]#, GPIO22, GPIO1, SERR#, STOP#, TRDY#<br>Interrupt Signals: PIRQ[D:A]#, PIRQ[H:E]#/GPIO[5:2] |
| VIH2/VIL2 | Clock Signals: CLK14, CLK48<br>Power Management Signals: MCH_SYNC#, THRM#, VRMPWRGD<br>GPIO Signals:<br>GPIO[39:37,33,7, 6]<br>Strap Signals: SPKR, SATALED# (Strap purposes only) |

**Table 8-92.DC Characteristic Input Signal Association (Sheet 2 of 2)**

| Symbol | Associated Signals |
|---|---|
| $V_{IH3}/V_{IL3}$ | Clock Signals: PCICLK<br>LPC/Firmware Hub Signals: LAD[3:0]/FWH[3:0], LDRQ0#, LDRQ1#/GPIO23<br>Power Management Signals:<br>LAN_RST#, BM_BUSY#/GPIO0, CLKRUN#, LAN_RST#<br>PCI Signals: PME#<br>Interrupt Signals: SERIRQ<br>CPU Signals: A20GATE, RCIN#<br>USB Signals: OC[4:0]#, OC[7:5]#/GPIO[31:29]<br>GPIO Signals: STRAP1#/GPIO48, STRAP2#/GPIO17, GPIO0<br>SPI Signals: SPI_CS#, SPI_MISO, SPI_ARB |
| $V_{IH4}/V_{IL4}$ | SMBus Signals: SMBCLK, SMBDATA<br>System Management Signals: SMBALERT#/GPIO11, SMLINK[1:0] |
| $V_{IH5}/V_{IL5}$ | LAN Signals: LAN_CLK, LAN_RXD[2:0]<br>EEPROM Signals: EE_DIN<br>Strap Signals: EE_CS, EE_DOUT (Strap purposes only) |
| $V_{IH6}/V_{IL6}$ | Processor Signals: FERR#, THRMTRIP#<br>**GPIO Signals**: GPIO49/CPUPWRGD |
| $V_{IMIN7}/V_{IMAX7}$ | PCI Express* Data RX Signals: PER[p,n][4:1] |
| $V_{IH8}/V_{IL8}$ | Real Time Clock Signals: RTCX1 |
| $V_{IMIN9}/V_{IMAX9}$ | SATA Signals:<br>SATA[1:0]RX[P,N] |
| $V_{IH10}/V_{IL10}$ | Strap Signals: HDA_SDOUT, HDA_SYNC (Strap purposes only)<br>**GPIO Signals**: GPIO34<br><br>See $V_{IL_HDA}/V_{IH_HDA}$ for High Definition Audio Low Voltage Mode (Netbook Only) |
| $V_{IH11}/V_{IL11}/$ $V_{cross(abs)}$ | Clock Signals: DMI_CLKN, DMI_CLKP, SATA_CLKN, SATA_CLKP |
| $V_{IH12}/V_{IL12}$ | Power Management Signals:<br>: BATLOW#, PWRBTN#, RI#, SYS_RESET#, WAKE#<br>System Management Signal: LINKALERT#<br>GPIO Signals: GPIO[28:26, 25:24, 15:12, 10:8] |
| $V_{IH13}/V_{IL13}$ | Power Management Signals: PWROK, RSMRST#, RTCRST#<br>System Management Signals: INTRUDER# |
| $V_{IH14}/V_{IL14}$ | Other Signals: INTVRMEN |
| $V_{DI}$ / $V_{CM}$ / $V_{SE}$ (5V Tolerant) | USB Signals: USBP[7:0][P,N] (Low-speed and Full-speed) |
| $V_{HSSQ}$ / $V_{HSDSC}$ / $V_{HSCM}$ (5V Tolerant) | USB Signals: USBP[7:0][P,N] (in High-speed Mode) |



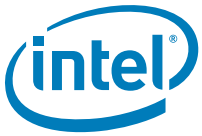
### Table 8-93.DC Input Characteristics (Sheet 1 of 2)

| Symbol | Parameter | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| $V_{IL1}$ | Input Low Voltage | –0.5 | 0.3(Vcc3_3) | V | |
| $V_{IH1}$ | Input High Voltage | 0.5(Vcc3_3) | V5REF + 0.5 | V | |
| $V_{IL2}$ | Input Low Voltage | –0.5 | 0.8 | V | |
| $V_{IH2}$ | Input High Voltage | 2.0 | Vcc3_3 + 0.5 | V | |
| $V_{IL3}$ | Input Low Voltage | –0.5 | 0.3(Vcc3_3) | V | |
| $V_{IH3}$ | Input High Voltage | 0.5(Vcc3_3) | Vcc3_3 + 0.5 | V | |
| $V_{IL4}$ | Input Low Voltage | –0.5 | 0.8 | V | |
| $V_{IH4}$ | Input High Voltage | 2.1 | VccSus3_3 + 0.5 | V | |
| $V_{IL5}$ | Input Low Voltage | -0.5 | 0.3(Vcc3_3) | V | |
| $V_{IH5}$ | Input High Voltage | 0.6(Vcc3_3) | Vcc3_3 + 0.5 | V | |
| $V_{IL6}$ | Input Low Voltage | –0.5 | 0.58(V_CPU_IO) | V | |
| $V_{IH6}$ | Input High Voltage | 0.73(V_CPU_IO) | V_CPU_IO + 0.5 | V | |
| $V_{IMIN7}$ | Minimum Input Voltage | 175 | | mVdiffp-p | 1 |
| $V_{IMAX7}$ | Maximum Input Voltage | | 1200 | mVdiffp-p | 1 |
| $V_{IL8}$ | Input Low Voltage | –0.5 | 0.10 | V | |
| $V_{IH8}$ | Input High Voltage | 0.40 | 1.2 | V | |
| $V_{IMIN9}$ | Minimum Input Voltage | 325 | | mVdiffp-p | 2 |
| $V_{IMAX9}$ | Maximum Input Voltage | | 600 | mVdiffp-p | 2 |
| $V_{IMIN9}$ | Minimum Input Voltage | 275 | | mVdiffp-p | 3 |
| $V_{IMAX9}$ | Maximum Input Voltage | | 750 | mVdiffp-p | 3 |
| $V_{IL10}$ | Input Low Voltage | –0.5 | 0.35(Vcc3_3) | V | |
| $V_{IH10}$ | Input High Voltage | 0.65(Vcc3_3) | Vcc3_3 + 0.5 | V | |
| $V_{IL11}$ | Input Low Voltage | -0.150 | 0.150 | V | |
| $V_{IH11}$ | Input High Voltage | 0.660 | 0.850 | V | |
| $V_{IL12}$ | Input Low Voltage | –0.5 | 0.8 | V | |
| $V_{IH12}$ | Input High Voltage | 2.0 | VccSus3_3 + 0.5 | V | |
| $V_{IL13}$ | Input Low Voltage | –0.5 | 0.78 | V | |
| $V_{IH13}$ | Input High Voltage | 2.0 | VccRTC + 0.5 | V | 4 |
| $V_{IL14}$ | Input Low Voltage | –0.5 | 0.65 | V | |
| $V_{IH14}$ | Input High Voltage | 2.0 | VccRTC + 0.5 | V | 4 |
| $V_{cross(abs)}$ | Absolute Crossing Point | 0.250 | 0.550 | V | |
| $V_{DI}$ | Differential Input Sensitivity | 0.2 | | V | 5, 6 |

**Table 8-93.DC Input Characteristics (Sheet 2 of 2)**

| Symbol | Parameter | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| $V_{CM}$ | Differential Common Mode Range | 0.8 | 2.5 | V | 6, 7 |
| $V_{SE}$ | Single-Ended Receiver Threshold | 0.8 | 2.0 | V | |
| $V_{HSSQ}$ | HS Squelch Detection Threshold | 100 | 150 | mV | |
| $V_{HSDSC}$ | HS Disconnect Detection Threshold | 525 | 625 | mV | |
| $V_{HSCM}$ | HS Data Signaling Common Mode Voltage Range | –50 | 500 | mV | 6 |
| $V_{HSSQ}$ | HS Squelch detection threshold | 100 | 150 | mV | |
| $V_{HSDSC}$ | HS disconnect detection threshold | 525 | 625 | mV | |
| $V_{HSCM}$ | HS data signaling common mode voltage range | –50 | 500 | mV | |

**NOTES:**
1. PCI Express mVdiff p-p = |PETp[x] – PETn[x]|
2. Applicable only when SATA port signaling rate is 1.5 Gb/s: SATA Vdiff, tx ($V_{IMAX/MIN10}$) is measured at the SATA connector on the transmit side (generally, the motherboard connector), where
   SATA mVdiff p-p = |SATA[x]TXP/RXP – SATA[x]TXN/RXN|
3. Applicable only when SATA port signaling rate is 3 Gb/s: SATA Vdiff, tx ($V_{IMAX/MIN10}$) is measured at the SATA connector on the transmit side (generally, the motherboard connector), where
   SATA mVdiff p-p = |SATA[x]TXP/RXP – SATA[x]TXN/RXN|
4. VccRTC is the voltage applied to the VccRTC well of Chipset. When the system is in a G3 state, this is generally supplied by the coin cell battery, but for S5 and greater, this is generally VccSus3_3.
5. $V_{DI}$ = | USBPx[P] – USBPx[N]
6. Applies to High-speed USB 2.0
7. Includes $V_{DI}$ range



### Table 8-94.DC Characteristic Output Signal Association

| Symbol | Associated Signals |
|---|---|
| $V_{OH1}/V_{OL1}$ | Processor Signals: A20M#, DPSLP#, DPRSTP#, IGNNE#, INIT#, INTR, NMI, SMI#, STPCLK#, CPUPWRGD/GPIO49 |
| $V_{OH2}/V_{OL2}$ | PCI Signals: AD[31:0], C/BE[3:0]#, DEVSEL#, FRAME#, IRDY#, PAR, PERR#, PLOCK#, SERR#, STOP#, TRDY#<br>Intel HD Audio Signals: HDA_RST#, HDA_SDOUT, ACZ_SYNC, HDA_BIT_CLK<br>**NOTE:** See $V_{OH_HDA}/V_{OL_HDA}$ for High Definition Audio Low Voltage Mode |
| $V_{OH3}/V_{OL3}$ | SMBus Signals: SMBCLK[1], SMBDATA[1]<br>System Management Signals: SMLINK[1:0][1]<br>**GPIO Signals:** GPIO11/SMBALERT[1] |
| $V_{OH4}/V_{OL4}$ | Power Management Signals: DPRSLPVR, PLTRST#, SLP_S3#, SLP_S4#, SLP_S5#, STP_CPU#, STP_PCI#, SUSCLK, SUS_STAT<br>GPIO Signals:GPIO[39, 38, 33, 20, 18, 16, 7, 6]<br>Other Signals: SPKR<br>SATA Signal: SATALED#<br>EEPROM Signals: EE_CS, EE_DOUT, EE_SHCLK |
| $V_{OH5}/V_{OL5}$ | **USB Signal**s: USBP[7:0][P,N] in Low-speed and Full-speed Modes |
| $V_{OMIN6}/V_{OMAX6}$ | **PCI Express* Data TX Signals:** PET[p,n][4:1] on Chipset. |
| $V_{OMIN7}/V_{OMAX7}$ | SATA Signals:<br>SATA[1:0]TX[P,N] |
| $V_{OH8}/V_{OL8}$ | **LPC/Firmware Hub Signals:** LAD[3:0]/FWH[3:0], LFRAME#/FWH[4]<br>**PCI Signals:** PCIRST#, GNT[2:1]#,STRAP0#, STRAP1#/GPIO48, STRAP0#/GPIO17<br>**GPIO Signals:**<br>: GPIO23/LDRQ1#, GPIO22, GPIO[5:2]/PIRQ[H:E]#, GPIO1, GPIO0/ BM_BUSY#, GPIO34<br>**Interrupt Signals:** SERIRQ<br>**SPI Signals:**<br>SPI_CS#, SPI_MOSI, SPI_CLK<br>**Processor Interface Signal:** INIT3_3V#<br>**LAN Signals:** LAN_RSTSYNC, LAN_TXD[2:0] |
| $V_{OH9}/V_{OL9}$ | GPIO Signals: GPIO[28:26, 25:24, 15:12, 10:8], GPIO[31:29]/OC[7:5]# |
| $V_{HSOI}$<br>$V_{HSOH}$<br>$V_{HSOL}$<br>$V_{CHIRPJ}$<br>$V_{CHIRPK}$ | USB Signals: USBP[7:0][P:N] in High-speed Mode |
| $V_{OH_HDA}/V_{OL_HDA}$ | Intel HD Audio Signals: HDA_RST#, HDA_SDOUT, HDA_SYNC |

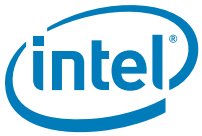
**NOTES:**
1. These signals are open drain

### Table 8-95.DC Output Characteristics

| Symbol | Parameter | Min | Max | Unit | $I_{OL}$ / $I_{OH}$ | Notes |
|---|---|---|---|---|---|---|
| $V_{OL1}$ | Output Low Voltage | | 0.255 | V | 3 mA | 1 |
| $V_{OH1}$ | Output High Voltage | V_CPU_IO – 0.3 | | V | -3 mA | |
| $V_{OL2}$ | Output Low Voltage | | 0.1(Vcc3_3) | V | 1.5 mA | |
| $V_{OH2}$ | Output High Voltage | 0.9(Vcc3_3) | | V | -0.5 mA | 2 |
| $V_{OL3}$ | Output Low Voltage | | 0.4 | V | 4 mA | |
| $V_{OH3}$ | Output High Voltage | VccSus3_3 – 0.5 | | V | -2 mA | 2 |
| $V_{OL4}$ | Output Low Voltage | | 0.4 | V | 6 mA | |
| $V_{OH4}$ | Output High Voltage | Vcc3_3 – 0.5 | | V | -2 mA | |
| $V_{OL5}$ | Output Low Voltage | | 0.4 | V | 5 mA | |
| $V_{OH5}$ | Output High Voltage | Vcc3_3 – 0.5 | | V | -2 mA | |
| $V_{OMIN6}$ | Minimum Output Voltage | 800 | | mVdif fp-p | | 3 |
| $V_{OMAX6}$ | Maximum Output Voltage | | 1200 | mVdif fp-p | | 3 |
| $V_{OMIN7}$ | Minimum Output Voltage | 400 | | mVdif fp-p | | 4 |
| $V_{OMAX7}$ | Maximum Output Voltage | | 600 | mVdif fp-p | | 4 |
| $V_{OMIN7}$ | Minimum Output Voltage | 400 | | mVdif fp-p | | 5 |
| $V_{OMAX7}$ | Maximum Output Voltage | | 700 | mVdif fp-p | | 5 |
| $V_{OL8}$ | Output Low Voltage | | 0.1(Vcc3_3) | V | 1.5 mA | |
| $V_{OH8}$ | Output High Voltage | 0.9(Vcc3_3) | | V | -0.5 mA | 2 |
| $V_{OL9}$ | Output Low Voltage | | 0.4 | V | 6 mA | |
| $V_{OH9}$ | Output High Voltage | VccSus3_3 – 0.5 | | V | -0.5 mA | |
| $V_{HSOI}$ | HS Idle Level | –10.0 | 10.0 | mV | | |
| $V_{HSOH}$ | HS Data Signaling High | 360 | 440 | mV | | |
| $V_{HSOL}$ | HS Data Signaling Low | –10.0 | 10.0 | mV | | |
| $V_{CHIRPJ}$ | Chirp J Level | 700 | 1100 | mV | | |
| $V_{CHIRPK}$ | Chirp K Level | –900 | –500 | mV | | |
| $V_{OL_HDA}$ | Output Low Voltage | | 0.1(VccHDA) | V | 1.5 mA | |
| $V_{OH_HDA}$ | Output High Voltage | 0.9(Vcc_HDA) | | V | -0.5 mA | |

**NOTES:**

1. Maximum $I_{ol}$ for CPUPWRGD is 12 mA for short durations (<500 mS per 1.5 s) and 9 mA for long durations.



2. The SERR#, PIRQ[H:A], GPIO11, SMBDATA, SMBCLK, LINKALERT#, and SMLINK[1:0] signal has an open drain driver and SATALED# has an open collector driver, and the $V_{OH}$ spec does not apply. This signal must have an external pull-up resistor.
3. PCI Express mVdiff p-p = |PETp[x] – PETn[x]|
4. Applicable only when SATA port signaling rate is 1.5 Gb/s: SATA Vdiff, TX ($V_{OMAX/MIN8}$) is measured at the SATA connector on the transmit side closest to Chipset (generally, the motherboard connector), where SATA mV diff p-p = |SATAxTXP – SATAxTXN|.
5. Applicable only when SATA port signaling rate is 3 Gb/s: SATA Vdiff, TX ($_{VOMAX/MIN8}$) is measured at the SATA connector on the transmit side closest to Chipset (generally, the motherboard connector), where SATA mV diff p-p = |SATAxTXP – SATAxTXN|.

### Table 8-96.Other DC Characteristics

| Symbol | Parameter | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| V5REF | Chipset Core Well Reference Voltage | 4.75 | 5.25 | V | 1 |
| Vcc3_3 | I/O Buffer Voltage | 3.135 | 3.465 | V | 1 |
| Vcc1_5, VccUSBPLL, VccSATAPLL, VccDMIPLL | Internal Logic and I/O Buffer Voltage | 1.425 | 1.575 | V | 1 |
| V_CPU_IO | Processor Interface | 0.945 | 1.25 | V | 1, 2 |
| V5REF_Sus | Suspend Well Reference Voltage | 4.75 | 5.25 | V | 1 |
| VccSus3_3 | Suspend Well I/O Buffer Voltage | 3.135 | 3.465 | V | 1 |
| Vcc1_05 | Internal Logic Voltage | 0.998 | 1.102 | V | 1 |
| VccRTC | Battery Voltage | 2.0 | 3.6 | V | 1 |
| $V_{DI}$ | Differential Input Sensitivity | 0.2 | | V | \|(USBPx+,USBPx-)\| |
| $V_{CM}$ | Differential Common Mode Range | 0.8 | 2.5 | V | Includes $V_{DI}$ |
| $V_{CRS}$ | Output Signal Crossover Voltage | 1.3 | 2.0 | V | |
| $V_{SE}$ | Single Ended Rcvr Threshold | 0.8 | 2.0 | V | |
| $I_{LI1}$ | ATA Input Leakage Current | –200 | 200 | µA | (0 V < $V_{IN}$ < 5 V) |
| $I_{LI2}$ | PCI_3V Hi-Z State Data Line Leakage | –10 | 10 | µA | (0 V < $V_{IN}$ < 3.3 V) |
| $I_{LI3}$ | PCI_5V Hi-Z State Data Line Leakage | –70 | 70 | µA | Max $V_{IN}$ = 2.7 V Min $V_{IN}$ = 0.5 V |
| $I_{LI4}$ | Input Leakage Current – Clock signals | –100 | +100 | µA | 1 |
| $C_{IN}$ | Input Capacitance – All Other | | 12 | pF | $F_C$ = 1 MHz |
| $C_{OUT}$ | Output Capacitance | | 12 | pF | $F_C$ = 1 MHz |
| $C_{I/O}$ | I/O Capacitance | | 12 | pF | $F_C$ = 1 MHz |

### Table 8-96.Other DC Characteristics

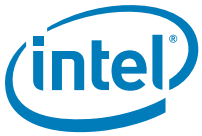
| Symbol | Parameter | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| | | Typical Value | | | |
| $C_L$ | XTAL1 | 6 | | pF | |
| $C_L$ | XTAL2 | 6 | | pF | |

**NOTES:**
1. For all noise components ≤20 MHz, the sum of the DC voltage and the AC noise component must be within the specified DC min/max operating range on the Chipset supply voltages.
2. The tolerances shown in Table 8-96 are inclusive of all noise from DC up to 20 MHz. In testing, the voltage rails should be measured with a bandwidth limited oscilloscope that has a roll off of 3 dB/decade above 20 MHz.
3. Includes CLK14, CLK48, LAN_CLK, and PCICLK.

## 8.4 AC Characteristics

### Table 8-97.Clock Timings (Sheet 1 of 2)

| Sym | Parameter | Min | Max | Unit | Notes | Figure |
|---|---|---|---|---|---|---|
| **PCI Clock (PCICLK)** | | | | | | |
| t1 | Period | 30 | 33.3 | ns | | 8-21 |
| t2 | High Time | 11 | | ns | | |
| t3 | Low Time | 11 | | ns | | |
| t4 | Rise Time | 1 | 4 | V/ns | | |
| t5 | Fall Time | 1 | 4 | V/ns | | |
| **14 MHz Clock (CLK14)** | | | | | | |
| t6 | Period | 67 | 70 | ns | | 8-21 |
| t7 | High Time | 20 | | ns | | |
| t8 | Low Time | 20 | | ns | | |
| t41 | Rising Edge Rate | 1.0 | 4.0 | V/ns | 1 | |
| t42 | Falling Edge Rate | 1.0 | 4.0 | V/ns | | |
| **48 MHz Clock (CLK48)** | | | | | | |
| $f_{clk48}$ | Operating Frequency | 48.000 | | MHz | 2 | |
| t9 | Frequency Tolerance | | 100 | ppm | | |
| t10 | High Time | 7 | | ns | | 8-21 |
| t11 | Low Time | 7 | | ns | | |
| t12 | Rise Time | | 1.2 | ns | | |
| t13 | Fall Time | | 1.2 | ns | | |



### Table 8-97.Clock Timings (Sheet 2 of 2)

| Sym | Parameter | Min | Max | Unit | Notes | Figure |
|---|---|---|---|---|---|---|
| **SMBus Clock (SMBCLK)** | | | | | | |
| $f_{smb}$ | Operating Frequency | 10 | 16 | KHz | | |
| t18 | High time | 4.0 | 50 | us | 3 | 8-30 |
| t19 | Low time | 4.7 | | us | | |
| t20 | Rise time | | 1000 | ns | | |
| t21 | Fall time | | 300 | ns | | |
| **HDA_BIT_CLK (Intel® High Definition Audio Mode)** | | | | | | |
| fHDA | Operating Frequency | 24.0 | | MHz | | |
| | Frequency Tolerance | — | 100 | ppm | | |
| t26a | Input Jitter (refer to Clock Chip Specification) | — | 300 | ppm | | |
| t27a | High Time (Measured at 0.75Vcc) | 18.75 | 22.91 | ns | | 8-21 |
| t28a | Low Time (Measured at 0.35Vcc) | 18.75 | 22.91 | ns | | |
| **SATA Clock (SATA_CLKP, SATA_CLKN) / DMI Clock (DMI_CLKP, DMI_CLKN)** | | | | | | |
| t36 | Period | 9.997 | 10.0533 | ns | | |
| t37 | Rise time | 175 | 700 | ps | | |
| t38 | Fall time | 175 | 700 | ps | | |
| **Suspend Clock (SUSCLK)** | | | | | | |
| fsusclk | Operating Frequency | 32 | | kHz | 4 | |
| t39 | High Time | 10 | — | us | | |
| t39a | Low Time | 10 | — | us | | |

**NOTES:**
1. CLK14 edge rates in a system as measured from 0.8 V to 2.0 V.
2. The CLK48 expects a 40/60% duty cycle.
3. The maximum high time (t18 Max) provides a simple method for devices to detect bus idle conditions.
4. SUSCLK duty cycle can range from 30% minimum to 70% maximum.

### Table 8-98.SATA Interface Timings

| Sym | Parameter | Min | Max | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| UI | Gen I Operating Data Period | 666.43 | 670.12 | ps | | |
| UI-2 | Gen II Operating Data Period (3Gb/s) | 333.21 | 335.06 | ps | | |
| t120 | Rise Time | 0.2 | 0.41 | UI | 1 | |
| t121 | Fall Time | 0.2 | 0.41 | UI | 2 | |
| t122 | TX differential skew | — | 20 | ps | | |

### Table 8-98.SATA Interface Timings

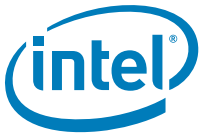
| Sym | Parameter | Min | Max | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| t123 | COMRESET | 310.4 | 329.6 | ns | 3 | |
| t124 | COMWAKE transmit spacing | 103.5 | 109.9 | ns | 3 | |
| t125 | OOB Operating Data period | 646.67 | 686.67 | ns | 4 | |

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

### Table 8-99.SMBus Timing

| Sym | Parameter | Min | Max | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| t130 | Bus Tree Time Between Stop and Start Condition | 4.7 | — | µs | | 8-30 |
| t131 | Hold Time after (repeated) Start Condition. After this period, the first clock is generated. | 4.0 | — | µs | | |
| t132 | Repeated Start Condition Setup Time | 4.7 | — | µs | | |
| t133 | Stop Condition Setup Time | 4.0 | — | µs | | |
| t134 | Data Hold Time | 0 | — | ns | 1 | |
| t135 | Data Setup Time | 250 | — | ns | | |
| t136 | Device Time Out | 25 | 35 | ms | 2 | |
| t137 | Cumulative Clock Low Extend Time (slave device) | — | 25 | ms | 3 | 8-31 |
| t138 | Cumulative Clock Low Extend Time (master device) | — | 10 | ms | 4 | |

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



### Table 8-100.Intel HD Audio Timing

| Sym | Parameter | Min | Max | Units | Notes | Figures |
|---|---|---|---|---|---|---|
| t143 | Time duration for which HDA_SDOUT is valid before HDA_BIT_CLK edge. | 7 | — | ns | 1 | 8-44 |
| t144 | Time duration for which HDA_SDOUT is valid after HDA_BIT_CLK edge. | 7 | — | ns | 1 | |
| t145 | Setup time for HDA_SDIN[2:0] at rising edge of HDA_BIT_CLK | 15 | — | ns | 1 | |
| t146 | Hold time for HDA_SDIN[2:0] at rising edge of HDA_BIT_CLK | 0 | — | ns | 1 | |

**NOTES:**
1. Audio link operating in Intel High Definition Audio mode.

### Table 8-101.LPC Timing

| Sym | Parameter | Min | Max | Units | Notes | Figures |
|---|---|---|---|---|---|---|
| t150 | LAD[3:0] Valid Delay from PCICLK Rising | 2 | 11 | ns | | 8-22 |
| t151 | LAD[3:0] Output Enable Delay from PCICLK Rising | 2 | — | ns | | 8-26 |
| t152 | LAD[3:0] Float Delay from PCICLK Rising | — | 28 | ns | | 8-24 |
| t153 | LAD[3:0] Setup Time to PCICLK Rising | 7 | — | ns | | 8-23 |
| t154 | LAD[3:0] Hold Time from PCICLK Rising | 0 | — | ns | | |
| t155 | LDRQ[1:0]# Setup Time to PCICLK Rising | 12 | — | ns | | |
| t156 | LDRQ[1:0]# Hold Time from PCICLK Rising | 0 | — | ns | | |
| t157 | LFRAME# Valid Delay from PCICLK Rising | 2 | 12 | ns | | 8-22 |

### Table 8-102.Miscellaneous Timings

| Sym | Parameter | Min | Max | Units | Notes | Figures |
|---|---|---|---|---|---|---|
| t160 | SERIRQ Setup Time to PCICLK Rising | 7 | — | ns | | 8-23 |
| t161 | SERIRQ Hold Time from PCICLK Rising | 0 | — | ns | | 8-23 |
| t162 | RI#, EXTSMI#, GPIO, USB Resume Pulse Width | 2 | — | RTCCLK | | 8-25 |
| t163 | SPKR Valid Delay from OSC Rising | — | 200 | ns | | 8-22 |
| t164 | SERR# Active to NMI Active | — | 200 | ns | | |
| t165 | IGNNE# Inactive from FERR# Inactive | — | 230 | ns | | |

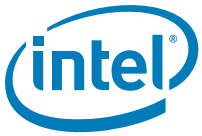
### Table 8-103.SPI Timings

| Sym | Parameter | Min | Max | Units | Notes | Figures |
|---|---|---|---|---|---|---|
| t180 | Serial Clock Frequency | 17.3 | 18.4 | MHz | 1 | |
| t182 | Duty cycle at the host | 40 | 60 | % | | 8-45 |
| t183 | Tco of SPI_MOSI with respect to serial clock falling edge at the host | -5 | 13 | ns | | |
| t184 | Setup of SPI_MISO with respect to serial clock falling edge at the host | 16 | — | ns | | |
| t185 | Hold of SPI_MISO with respect to serial clock falling edge at the host | 0 | — | ns | | |
| t186 | Setup of SPI_CS# assertion with respect to serial clock rising at the host. | 30 | — | ns | | |
| t187 | Hold of SPI_CS# deassertion with respect to serial clock falling at the host. | 30 | — | ns | | |

**NOTES:**
1. The typical clock frequency driven by Chipset is 17.9 MHz.

### Table 8-104.(Power Sequencing and Reset Signal Timings (Sheet 1 of 2)

| Sym | Parameter | Min | Max | Units | Notes | Fig |
|---|---|---|---|---|---|---|
| t200 | VccRTC active to RTCRST# inactive | 18 | — | ms | | 8-32<br>8-33 |
| t201 | V5REF_Sus active to VccSus3_3 active | 0 | — | ms | 1 | |
| t202 | VccSus3_3 active to VccSus1_05 active | — | — | — | 2 | |
| t203 | VccRTC supply active to VccSus supplies active | 0 | — | ms | 3 | |
| t204 | VccSus supplies active to LAN_RST# inactive, RSMRST# inactive<br>(Nettop Only) | 10 | — | ms | | 8-32<br>8-34 |
| t205 | VccSus supplies active to RSMRST# inactive<br>(Netbook Only) | 5 | — | ms | | 8-33<br>8-35 |
| t208 | VccSUS supplies active to LAN_RST# inactive<br>(Netbook Only) | 10 | — | ms | | 8-33 |
| t209 | V5REF active to Vcc3_3 active | 0 | — | ms | 1 | 8-32<br>8-33 |
| t211 | Vcc1_5 active to V_CPU_IO active | — | — | — | 1 | |
| t212 | VccSUS supplies active to Vcc supplies active<br>(Netbook Only) | 0 | — | ms | 5 | 8-33 |
| t213 | VccSus supplies active to Vcc supplies active<br>(Nettop Only) | 0 | — | ms | 3 | 8-32 |



### Table 8-104.(Power Sequencing and Reset Signal Timings (Sheet 2 of 2)

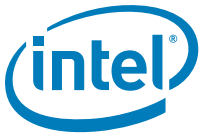
| Sym | Parameter | Min | Max | Units | Notes | Fig |
|---|---|---|---|---|---|---|
| t214 | Vcc supplies active to PWROK<br><br>**NOTE:** PWROK assertion indicates that PCICLK has been stable for at least 1 ms. | 99 | — | ms | 5 | 8-32<br>8-33<br>8-34<br>8-35<br>8-37<br>8-38<br>8-39<br>8-40 |
| t215 | V_CPU_IO active to STPCLK# and CPUSLP# inactive<br>(Nettop Only) | — | 50 | ns | | 8-34<br>8-37<br>8-38 |
| t216 | Vcc active to DPRSLPVR inactive and STPCLK#, STP_CPU#, STP_PCI#, DPSLP#, DPRSTP# inactive<br>(Netbook Only) | — | 50 | ns | | 8-35<br>8-39<br>8-40 |
| t217 | PWROK active and SYS_RESET# inactive to SUS_STAT# inactive and Processor I/F signals latched to strap value | 32 | 38 | RTCCLK | 6, 7 | 8-34<br>8-35<br>8-37<br>8-38<br>8-39<br>8-40 |
| t218 | SUS_STAT# inactive to PLTRST# inactive | 2 | 3 | RTCCLK | 7 | 8-34<br>8-35<br>8-37<br>8-38<br>8-39<br>8-40 |
| t228 | HDA_RST# active low pulse width | 1 | — | us | | |
| t229 | HDA_RST# inactive to HDA_BIT_CLK startup delay | 162.8 | — | ns | | |

**NOTES:**

1. 5REF must be powered up before Vcc3_3, or after Vcc3_3 within 0.7 V. Also, V5REF must power down after Vcc3_3, or before Vcc3_3 within 0.7 V.
2. The associated 3.3 V and 1.05 V supplies are assumed to power up or down ‘together’. If the integrated VccSus1_05 voltage regulator is not used: a) VccSus3_3 must power up before VccSus1_05 or after VccSus1_05 within 0.7 V, b) VccSus1_05 must power down before VccSus3_3 or after VccSus3_3 within 0.7 V.
3. The VccSus supplies must not be active while the VccRTC supply is inactive.
4. Vcc1_5 must power up before V_CPU_IO or after V_CPU_IO within 0.7 V, b) V_CPU_IO must power down before Vcc1_5 or after Vcc1_5 within 0.7 V.
5. Vcc supplies refer to all “core well” supplies: Vcc3_3, Vcc1_05, Vcc1_5, V5REF, VccUSBPLL, VccDMIPLL, VccSATAPLL, V_CPU_IO and VccHDA (Netbook Only). It implies that all “suspend wells” and VccRTC are stable too.
6. INIT# value determined by value of the CPU BIST Enable bit (Chipset Configuration Register Offset 3414h: bit 2).
7. These transitions are clocked off the internal RTC. 1 RTC clock is approximately from 28.992 µs to 32.044 µs.

### Table 8-105.Power Management Timings (Sheet 1 of 3)

| Sym | Parameter | Min | Max | Units | Notes | Fig |
|---|---|---|---|---|---|---|
| t230 | VccSus active to SLP_S5#, SLP_S4#, SLP_S3#, SUS_STAT#, PLTRST# and PCIRST#active | | 50 | ns | | 8-34<br>8-35 |
| t231<br>t232 | RSMRST# inactive to SUSCLK running, SLP_S5# inactive | | 110 | ms | 1, 2 | 8-34<br>8-35 |
| t233 | SLP_S5# inactive to SLP_S4# inactive | See Note Below | | | 3 | 8-34<br>8-35 |
| t234 | SLP_S4# inactive to SLP_S3# inactive | 1 | 2 | RTCCLK | 4 | 8-34<br>8-35 |
| t250 | Processor I/F signals latched prior to STPCLK# active<br>(Netbook Only) | 0 | | | 5 | 8-41<br>8-43 |
| t253 | DPSLP# active to STP_CPU# active<br>(Netbook Only) | 1 | 1 | PCICLK | 6 | 8-42<br>8-43 |
| t254 | STP_CPU# active to processor clock stopped<br>(Netbook Only) | 0 | – | PCICLK | 6, 7 | 8-43 |
| t255 | STP_CPU# active to DPRSTP#, DPRSLPVR active<br>(Netbook Only) | 0 | | | | 8-43 |
| t265 | Break Event to DPRSTP#, DPRSLPVR inactive<br>(C4 Exit)<br>(Netbook Only) | 1.5 | 1.8 | µs | 8 | 8-43 |
| t266 | DPRSLPVR, DPRSTP# inactive to STP_CPU# inactive and CPU Vcc ramped<br>(Netbook Only) | Programable. See D31:F0:AA, bits 3:2 | | µs | | 8-43 |
| t267 | Break Event to STP_CPU# inactive<br>(C3 Exit)<br>(Netbook Only) | 6 | Note 14 | PCICLK | 6, 9, 10 | 8-42 |
| t268 | STP_CPU# inactive to processor clock running<br>(Netbook Only) | 0 | 3 | PCICLK | 6, 7 | 8-43 |
| t269 | STP_CPU# inactive to DPSLP# inactive<br>(Netbook Only) | 1 | 1 | PCICLK | 6, 11 | 8-42<br>8-43 |
| t271 | S1 Wake Event to CPUSLP# inactive<br>(Nettop Only) | 1 | 25 | PCICLK | 6 | 8-36 |
| t273 | Break Event to STPCLK# inactive<br>(C2 Exit)<br>(Netbook Only) | 0 | | ns | | 8-41 |
| t274 | STPCLK# inactive to processor I/F signals unlatched<br>(Netbook Only) | 8 | 9 | PCICLK | 5, 6 | 8-41<br>8-43 |



### Table 8-105.Power Management Timings (Sheet 2 of 3)

| Sym | Parameter | Min | Max | Units | Notes | Fig |
|---|---|---|---|---|---|---|
| t280 | STPCLK# active to DMI Message | 0 | | PCICLK | 12 | 8-36<br>8-37<br>8-38<br>8-39<br>8-40 |
| t281 | DMI Message to CPUSLP# active<br>(Nettop Only) | 60 | 63 | PCICLK | 6 | 8-36 |
| t283 | DMI Message to SUS_STAT# active | 2 | | RTCCLK | 4 | 8-37<br>8-38<br>8-39<br>8-40 |
| t284 | SUS_STAT# active to PLTRST#, PCIRST# active<br>(Nettop Only) | 7 | 17 | RTCCLK | 4 | 8-37<br>8-38 |
| t285 | SUS_STAT# active to STP_PCI# active<br>(Netbook Only) | 2 | 10 | RTCCLK | 4 | 8-39<br>8-40 |
| t286 | STP_PCI# active to PLTRST# and PCIRST# active<br>(Netbook Only) | 5 | 7 | RTCCLK | 4 | 8-39<br>8-40 |
| t287 | PLTRST#, PCIRST# active to SLP_S3# active | 1 | 2 | RTCCLK | 4 | 8-37<br>8-38<br>8-39<br>8-40 |
| t288 | ($S3_{COLD}$ Configuration Only) SLP_S3# active to PWROK, VRMPWRGD inactive<br>(Netbook Only) | 0 | | ms | 13 | 8-39 |
| t289 | SLP_S3# active to PWROK, VRMPWRGD inactive<br>(Nettop Only) | 0 | | ms | 13 | 8-37 |
| t290 | ($S3_{COLD}$ Configuration Only) PWROK, VRMPWRGD inactive to Vcc supplies inactive<br>(Netbook Only) | 20 | | ns | 14, 15 | 8-39 |
| t291 | SLP_S3# active to SLP_S4# active | 1 | 2 | RTCCLK | 4 | 8-37<br>8-38<br>8-39<br>8-40 |
| t292 | ($S3_{HOT}$ Configuration Only) SLP_S4# active to VRMPWRGD and PWROK inactive | 0 | | ms | 13 | 8-38<br>8-40 |
| t293 | ($S3_{HOT}$ Configuration Only) PWROK, VRMPWRGD inactive to Vcc supplies inactive | 20 | | ns | 14, 15 | 8-38<br>8-40 |
| t294 | PWROK, VRMPWRGD inactive to Vcc supplies inactive<br>(Nettop Only) | 20 | | ns | 14, 15 | 8-37 |

### Table 8-105.Power Management Timings (Sheet 3 of 3)

| Sym | Parameter | Min | Max | Units | Notes | Fig |
|---|---|---|---|---|---|---|
| t295 | SLP_S4# active to SLP_S5# active | 1 | 2 | RTCCLK | 4, 16 | 8-37 8-38 8-39 8-40 |
| t296 | Wake Event to SLP_S5# inactive | 1 | 10 | RTCCLK | 4 | 8-37 8-38 8-39 8-40 |
| t297 | SLP_S5# inactive to SLP_S4# inactive | See Note Below | | | 3 | 8-37 8-38 8-39 8-40 |
| t298 | SLP_S4# inactive to SLP_S3# inactive | 1 | 2 | RTCCLK | 4 | 8-37 8-38 8-39 8-40 |
| t299 | S4 Wake Event to SLP_S4# inactive (S4 Wake) | See Note Below | | | 3 | 8-37 8-38 8-39 8-40 |
| t300 | S3 Wake Event to SLP_S3# inactive (S3 Wake) | 0 | small as possible | RTCCLK | 4 | 8-37 8-38 8-39 8-40 |
| t301 | CPUSLP# inactive to STPCLK# inactive (Nettop Only) | 8 | | PCICLK | | 8-36 |
| t302 | SLP_S3# inactive to Chipset check for PWROK active | 4 | 5 | msec | | 8-37 8-38 8-39 8-40 |
| t303 | SLP_S3# active to Vcc supplies inactive | 5 | | us | 15, 17 | |
| | **Other Timings** | | | | | |
| t310 | THRMTRIP# active to SLP_S3#, SLP_S4#, SLP_S5# active | | 3 | PCI CLK | | |
| t311 | RSMRST# rising edge transition from 20% to 80% | | 50 | us | | |
| t312 | RSMRST# falling edge transition | | | | 18 | |

**NOTES:**

1. If there is no RTC battery in the system, so VccRTC and the VccSus supplies come up together, the delay from RTCRST# and RSMRST# inactive to SUSCLK toggling may be as much as 2.5 s.
2. If the AFTERG3_EN bit (GEN_PMCON_3 Configuration Register Bit 1) is set to a 1, SLP_S5# will not be de-asserted until a wake event is detected. If the AFTERG3_EN bit is set to 0, SLP_S5# will deassert within the specification listed in the table.
3. The Min/Max times depend on the programming of the “SLP_S4# Minimum Assertion Width” and the “SLP_S4# Assertion Stretch Enable bits (D31:F0:A4h bits 5:3).
4. These transitions are clocked off the internal RTC. 1 RTC clock is approximately 28.992 µs to 32.044 µs.

5. Note that this does not apply for synchronous SMIs.
6. These transitions are clocked off the 33 MHz PCICLK. 1 PCICLK is approximately 30 ns.
7. This is a clock generator specification.
8. This is non-zero to enforce the minimum assert time for DPRSLPVR. If the minimum assert time for DPRSLPVR has been met, then this is permitted to be 0.
9. This is non-zero to enforce the minimum assert time for STP_CPU#. If the minimum assert time for STP_CPU# has been met, then this is permitted to be 0.
10. This value should be at most a few clocks greater than the minimum.
11. This value is programmable in multiples of 1024 PCI CLKs. Maximum is 8192 PCI CLKs (245.6 µs).
12. Chipset STPCLK# assertion will trigger the processor to send a stop grant acknowledge cycle. The timing for this cycle getting to Chipset is dependant on the processor and the memory controller.
13. Chipset has no maximum timing requirement for this transition. It is up to the system designer to determine if the SLP_S3#, SLP_S4# and SLP_S5# signals are used to control the power planes.
14. t290, t293, and t294 apply during S0 to G3 transitions only. In addition, the timings are not applied to V5REF. V5REF timings are bonded by power sequencing.
15. A Vcc supply is inactive when the voltage is below the min value specified in Table 8-96.
16. f the transition to S5 is due to Power Button Override, SLP_S3#, SLP_S4# and SLP_S5# are asserted together similar to timing t287 (PCIRST# active to SLP_S3# active).
17. t303 applies during S0 to S3-S5 transitions.
18. RSMRST# falling edge must transition to 0.8 V or less before VccSus3_3 drops to 2.1 V.

## 8.5 Timing Diagrams

**Figure 8-21. Clock Timing**

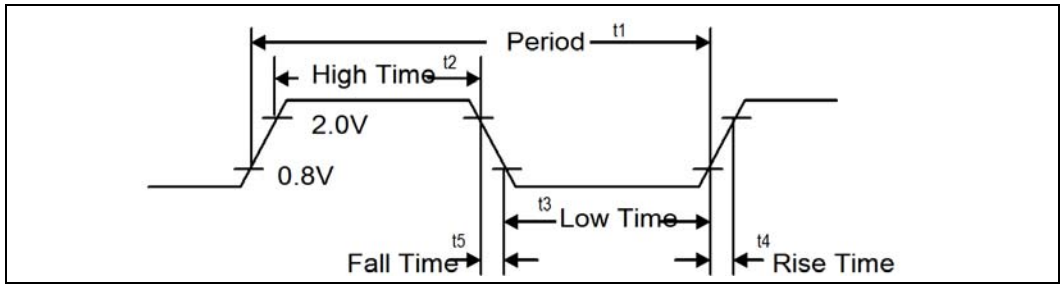



**Figure 8-22. Valid Delay from Rising Clock Edge**

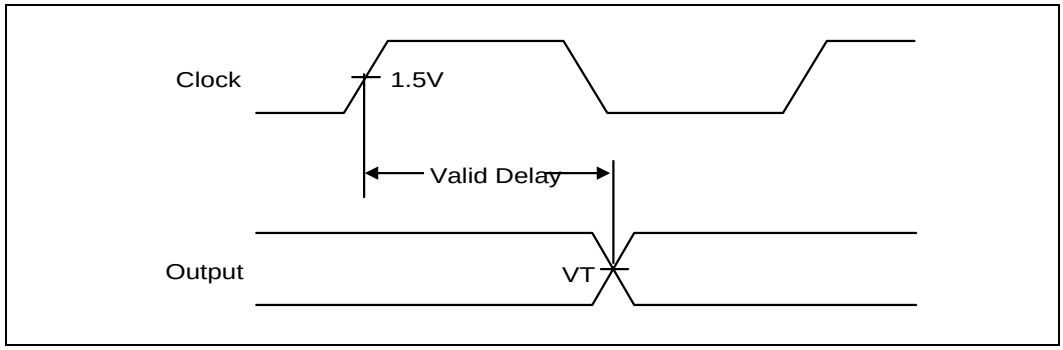


**Figure 8-23. Setup and Hold Times**

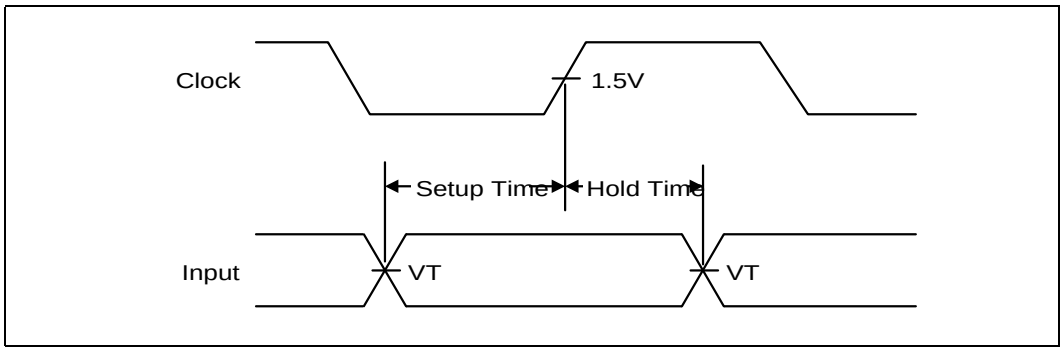



**Figure 8-24. Float Delay**

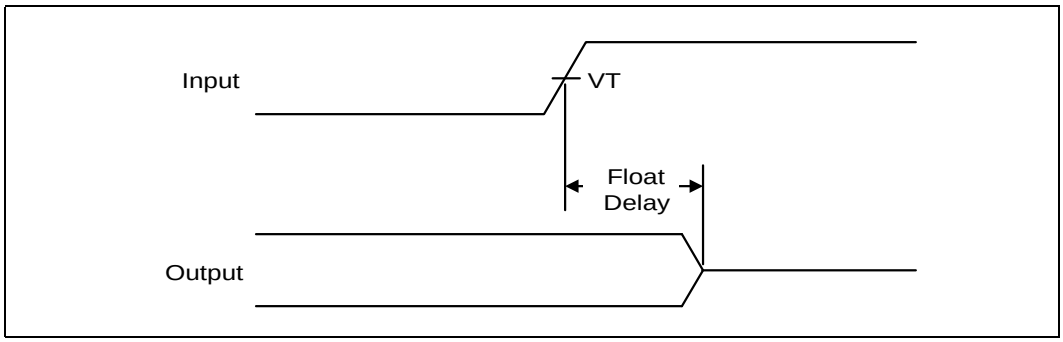



**Figure 8-25. Pulse Width**

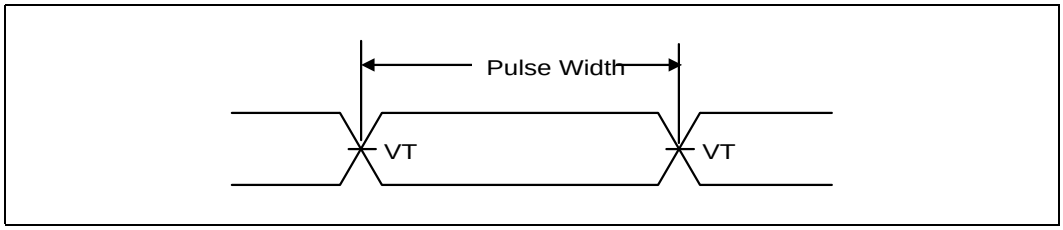



**Figure 8-26. Output Enable Delay**

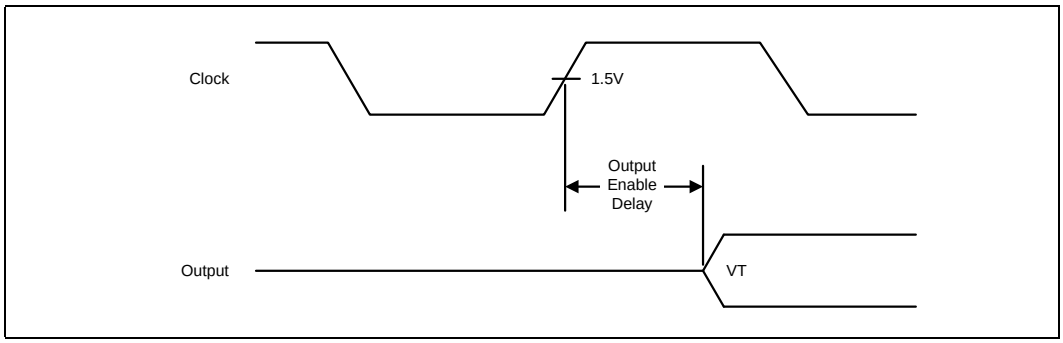


**Figure 8-27. USB Rise and Fall Times**

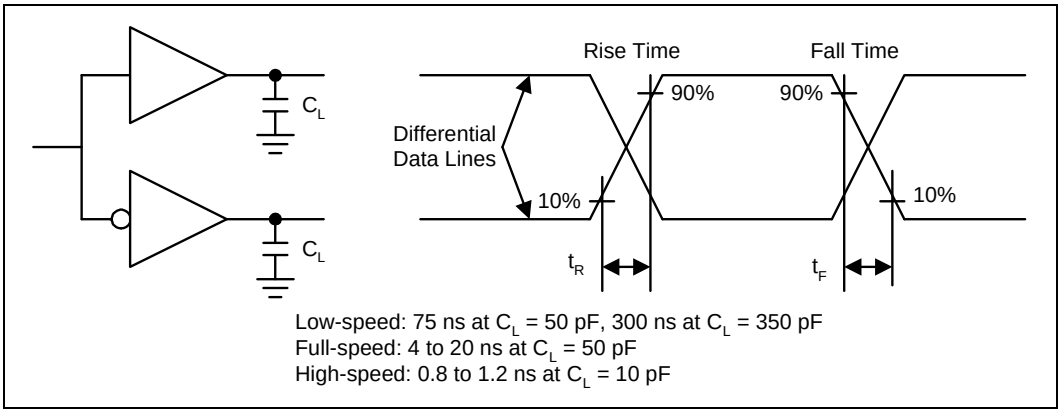



**Figure 8-28. USB Jitter**

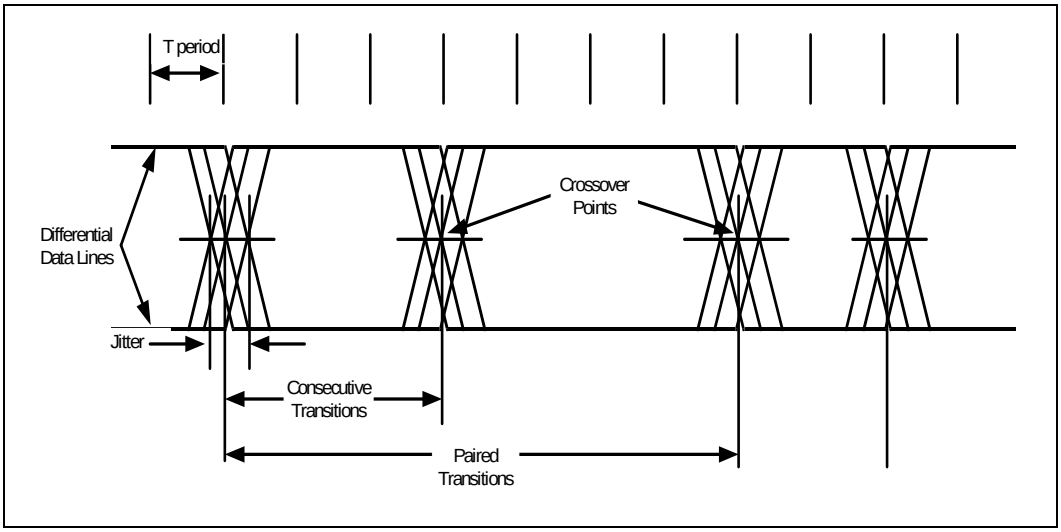



**Figure 8-29. USB EOP Width**

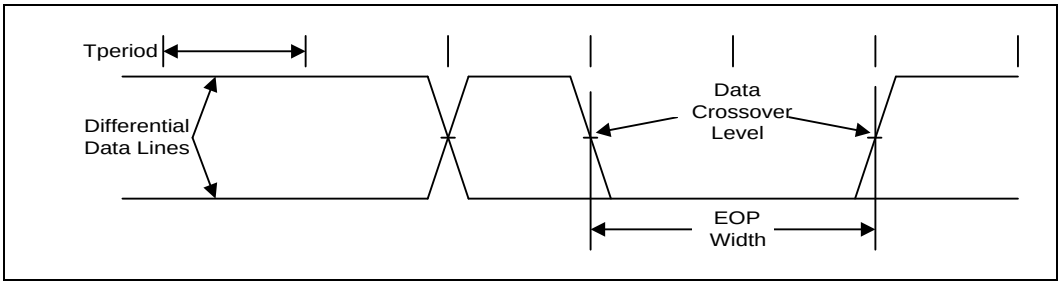


**Figure 8-30. SMBus Transaction**

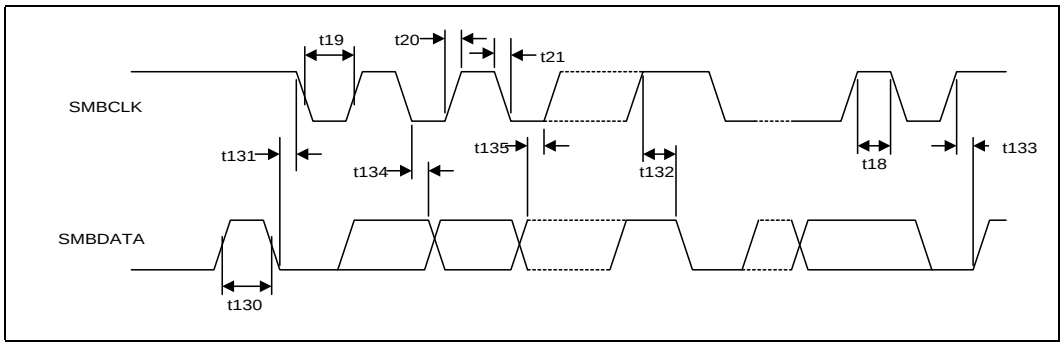
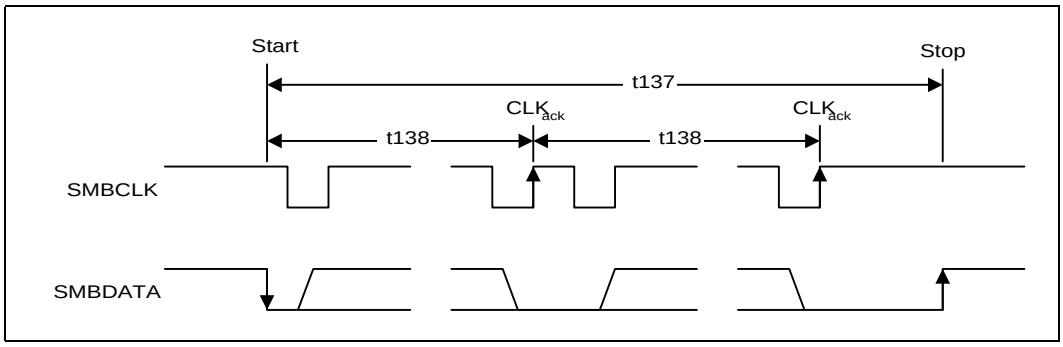
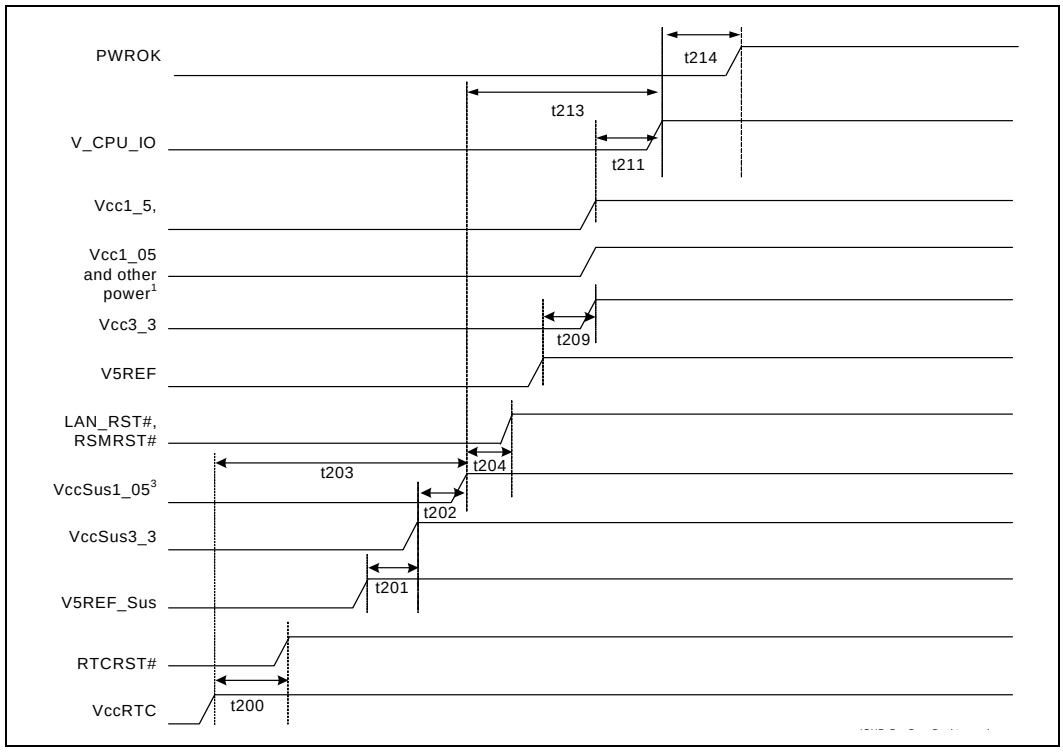



**Figure 8-31. SMBus Timeout**



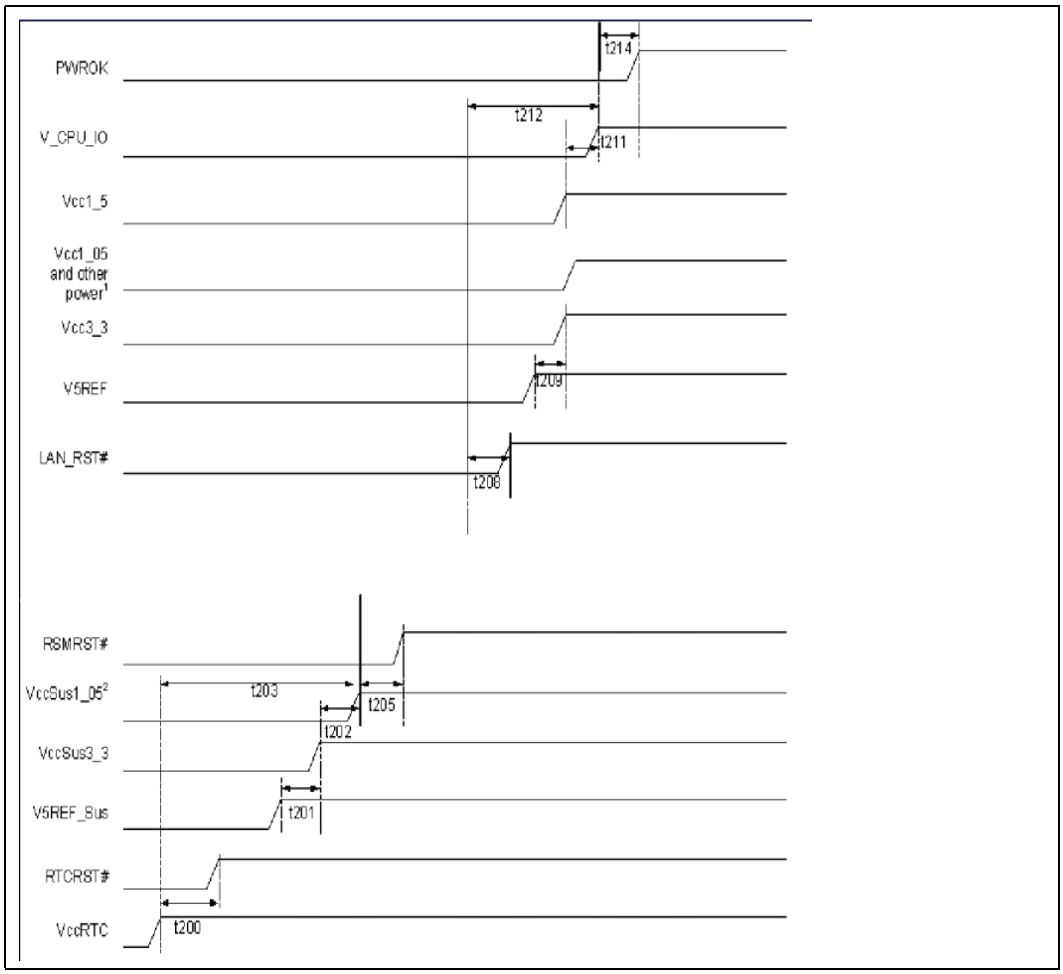
## Figure 8-32. Power Sequencing and Reset Signal Timings (Nettop Only)




**NOTES:**

1. Other power includes VccUSBPLL, VccDMIPLL, and VccSATAPLL. All of these power signals must independently meet the timings shown in the figure. There are no timing interdependencies between Vcc1_05 and these other power signals. There are also no timing interdependencies for these power signals, including Vcc1_05, to Vcc3_3 and Vcc1_5. However, If Vcc3_3 (core well buffer) is powered before Vcc1_05 (core well logic), core well signal states are indeterminate, undefined, and may glitch prior to PWROK assertion. Refer to Section 3.2 and Section 3.3 for a list of signals that will be determinate before PWROK.
2. PRWOK must not glitch, even if RSMRST# is low.
3. This power is supply by Chipset internal VR.

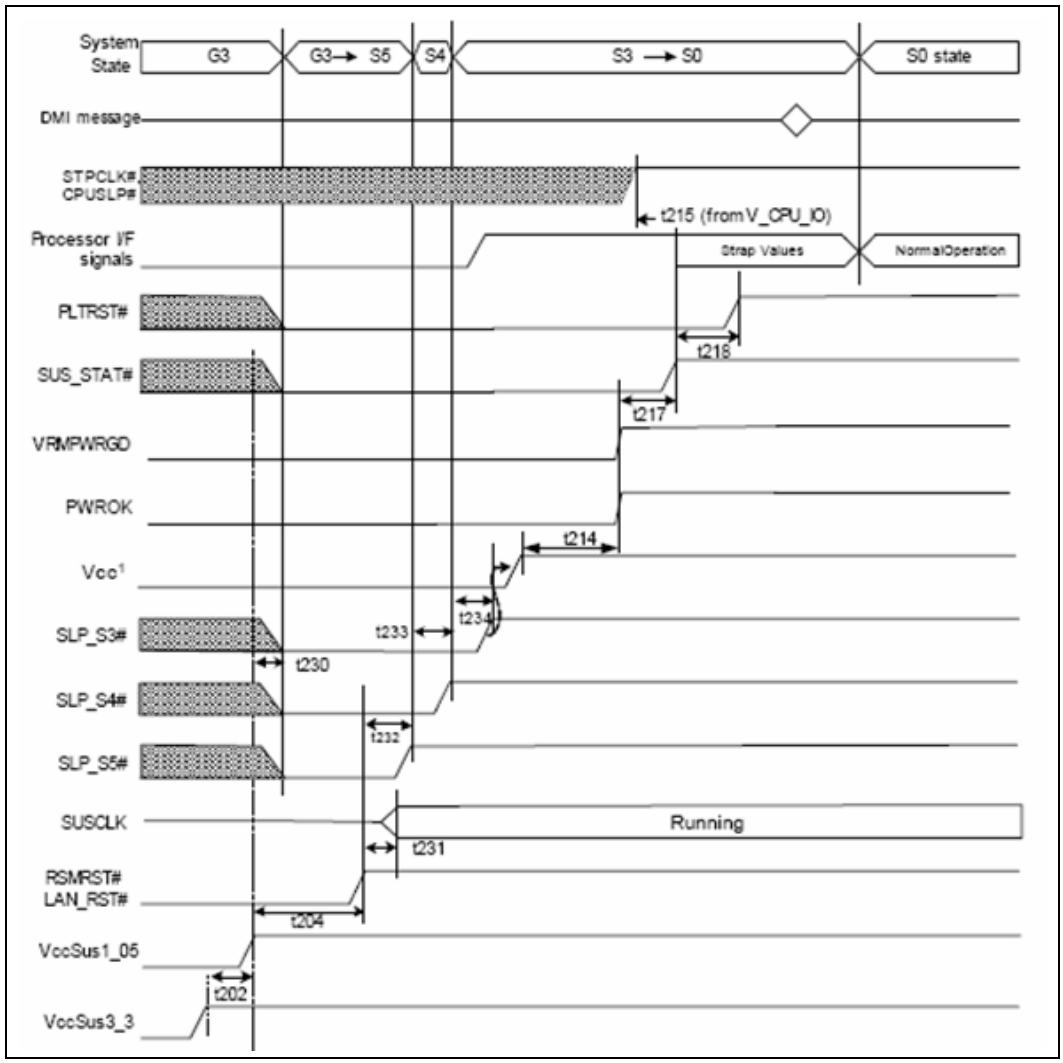
## Figure 8-33. Power Sequencing and Reset Signal Timings (Netbook Only)




**NOTES:**

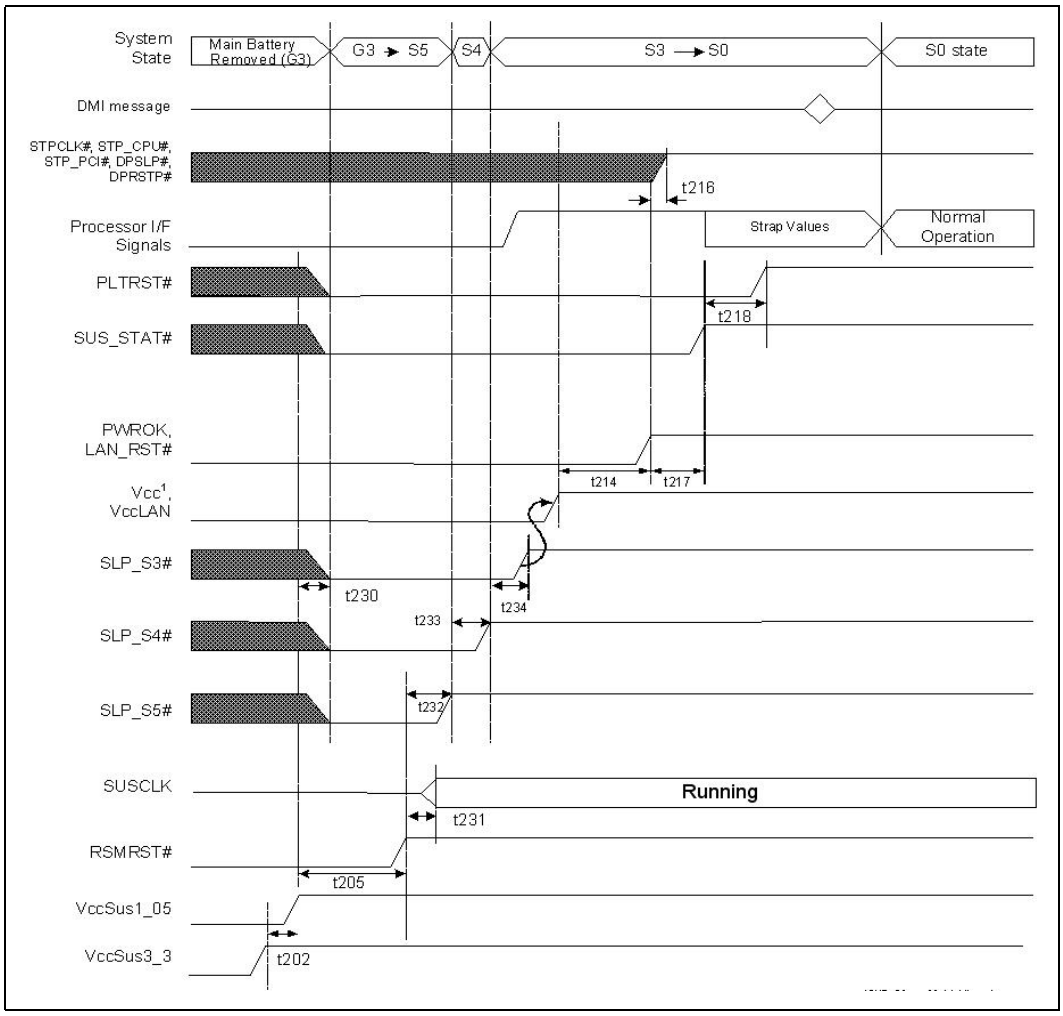
1. Other power includes VccUSBPLL, VccDMIPLL, and VccSATAPLL. All of these power signals must independently meet the timings shown in the figure. There are no timing interdependencies between Vcc1_05 and these other power signals. There are also no timing interdependencies for these power signals, including Vcc1_05, to Vcc3_3 and Vcc1_5. However, If Vcc3_3 (core well buffer) is powered before Vcc1_05 (core well logic), core well signal states are indeterminate, undefined, and may glitch prior to PWROK assertion. Refer to Section 3.2 and Section 3.3 for a list of signals that will be determinate before PWROK.
2. This power is supply by Chipset internal VR.

## Figure 8-34. G3 (Mechanical Off) to S0 Timings (Nettop Only)




**NOTES:**
1. Vcc includes Vcc1_5, Vcc3_3, Vcc1_05, VccUSBPLL, VccDMIPLL, VccSATAPLL, and V5REF.

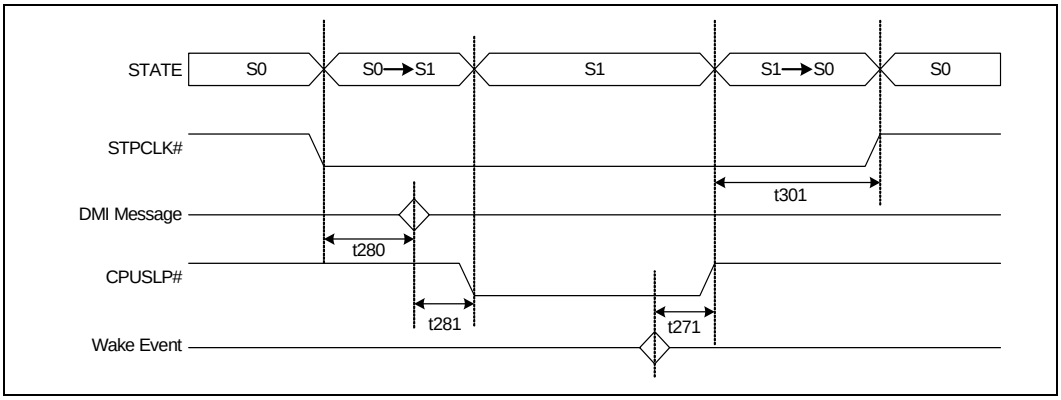
**Figure 8-35. G3 (Mechanical Off) to S0 Timings (Netbook Only)**



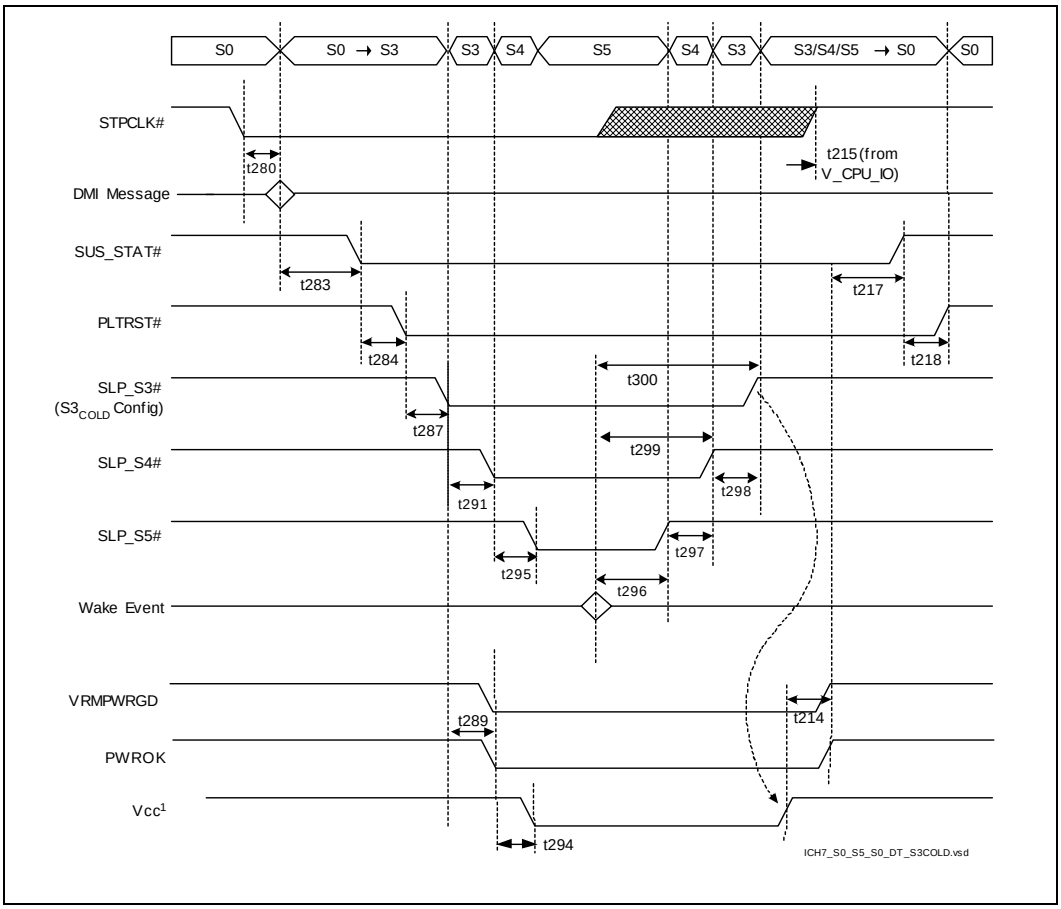

**NOTES:**

1. Vcc includes Vcc1_5,Vcc3_3, Vcc1_05, VccUSBPLL, VccDMIPLL, VccSATAPLL, and V5REF.

### Figure 8-36. S0 to S1 to S0 Timing (Nettop Only)



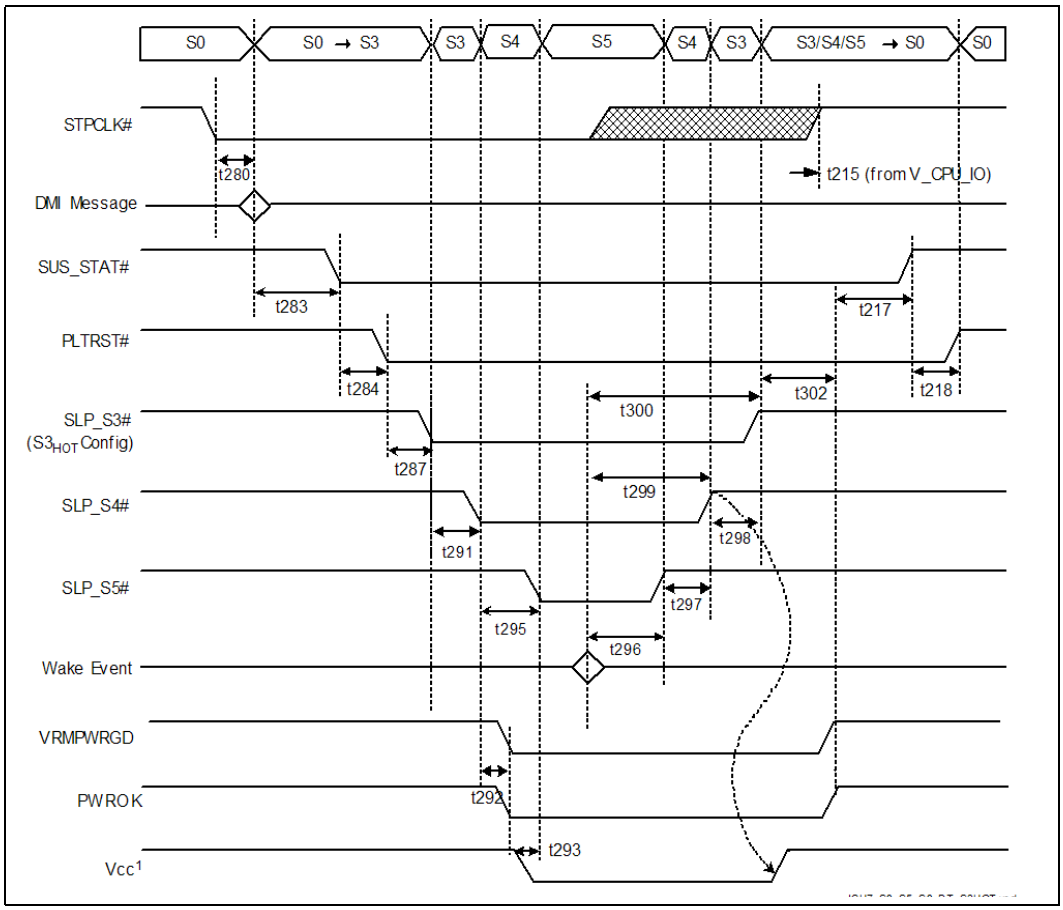

### Figure 8-37. S0 to S5 to S0 Timings, $S3_{COLD}$ (Nettop Only)




**NOTES:**

1. Vcc includes Vcc1_5, Vcc3_3, Vcc1_05, VccUSBPLL, VccDMIPLL, VccSATAPLL, and V5REF.
2. t294 is applicable when the system transitions from S0 to G3 only.

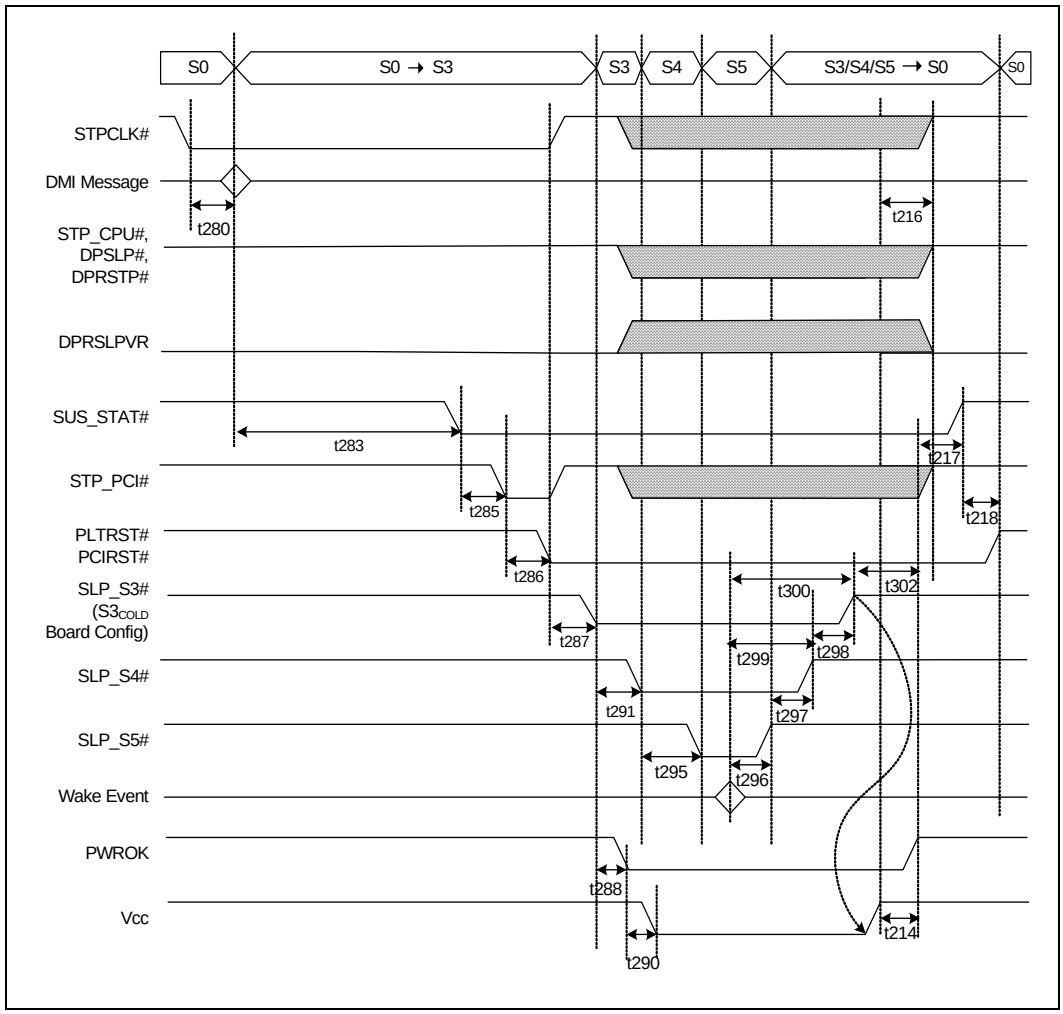
### Figure 8-38. S0 to S5 to S0 Timings, $S3_{HOT}$ (Nettop Only)




**NOTES:**

1. Vcc includes Vcc1_5, Vcc3_3, Vcc1_05, VccUSBPLL, VccDMIPLL, VccSATAPLL, and V5REF.
2. t293 is applicable when the system transitions from S0 to G3 only.

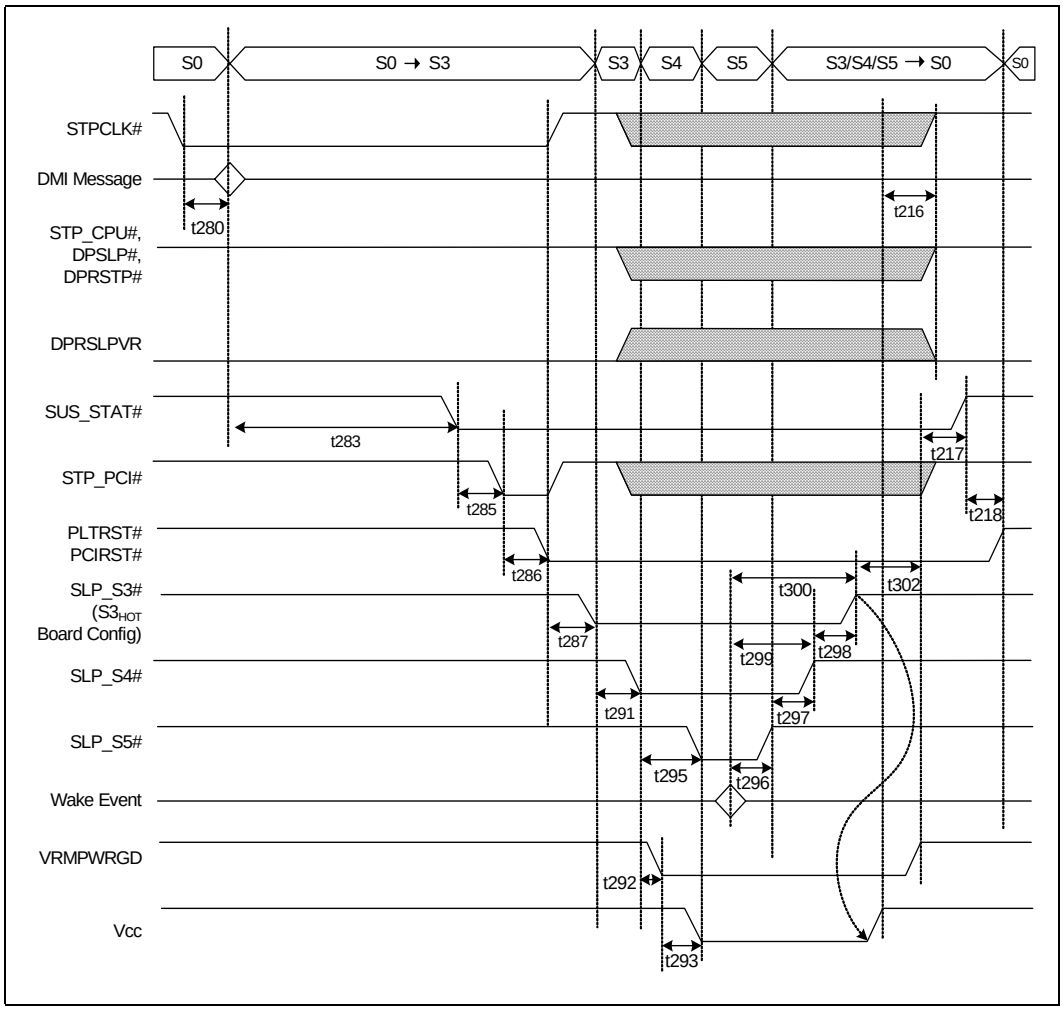
### Figure 8-39. S0 to S5 to S0 Timings, $S3_{COLD}$ (Netbook Only)




**NOTES:**

1. t290 is applicable when the system transitions from S0 to G3 only.
2. Vcc includes Vcc1_5, Vcc1_5_B, Vcc3_3, Vcc1_05, VccUSBPLL, VccDMIPLL, VccSATAPLL, and V5REF.

### Figure 8-40. S0 to S5 to S0 Timings, $S3_{HOT}$ (Netbook Only)




**NOTES:**
1. t293 is applicable when the system transitions from S0 to G3 only.
2. Vcc includes Vcc1_5, Vcc1_5_B, Vcc3_3, Vcc1_05, VccUSBPLL, VccDMIPLL, VccSATAPLL, and V5REF.

**Figure 8-41. C0 to C2 to C0 Timings (Netbook Only)**

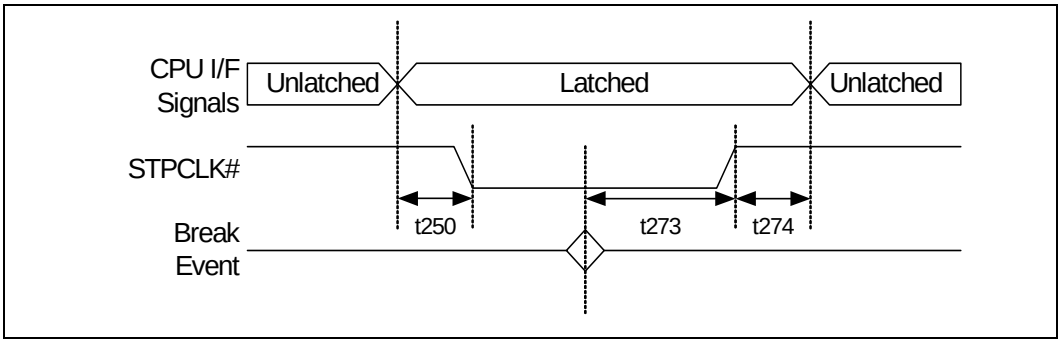



**Figure 8-42. C0 to C3 to C0 Timings (Nettop Only)**

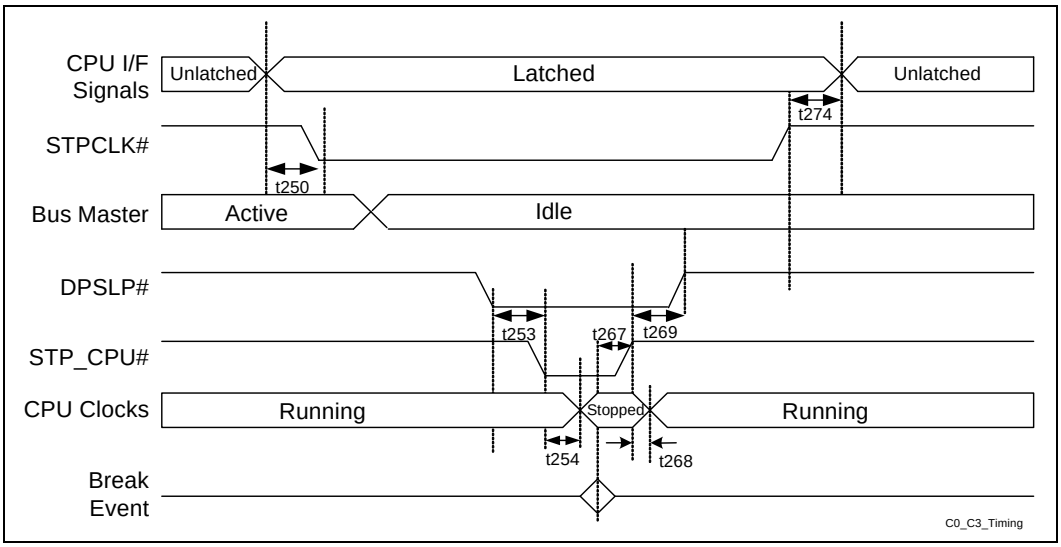


**Figure 8-43. C0 to C4 to C0 Timings (Netbook Only)**

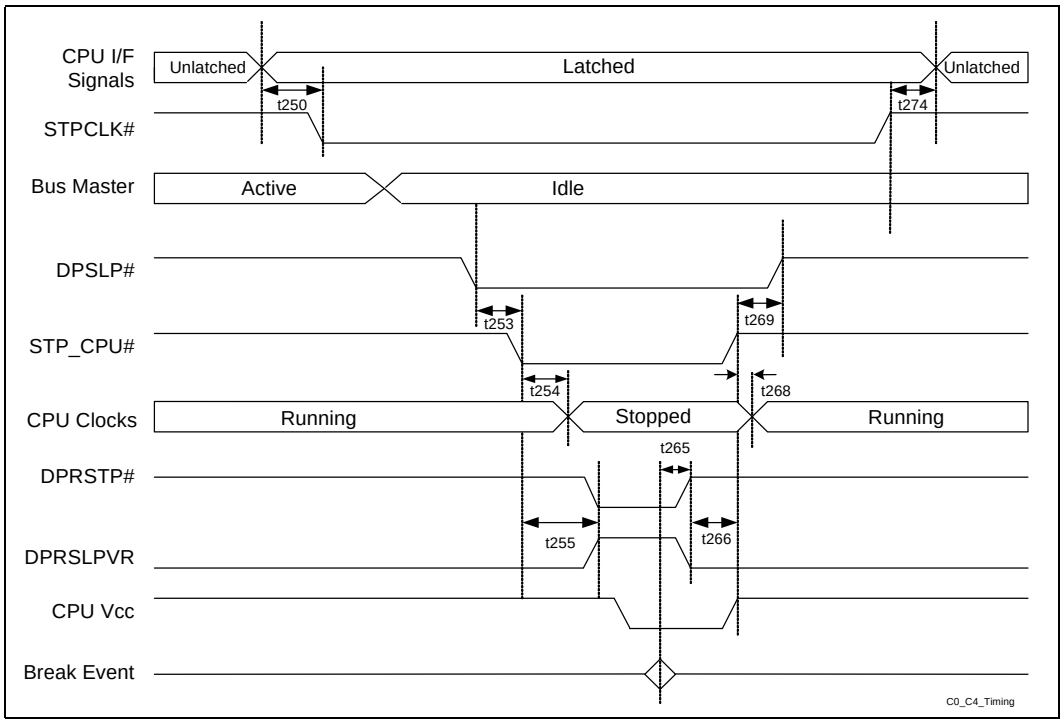



**Figure 8-44. Intel HD Audio Input and Output Timings**

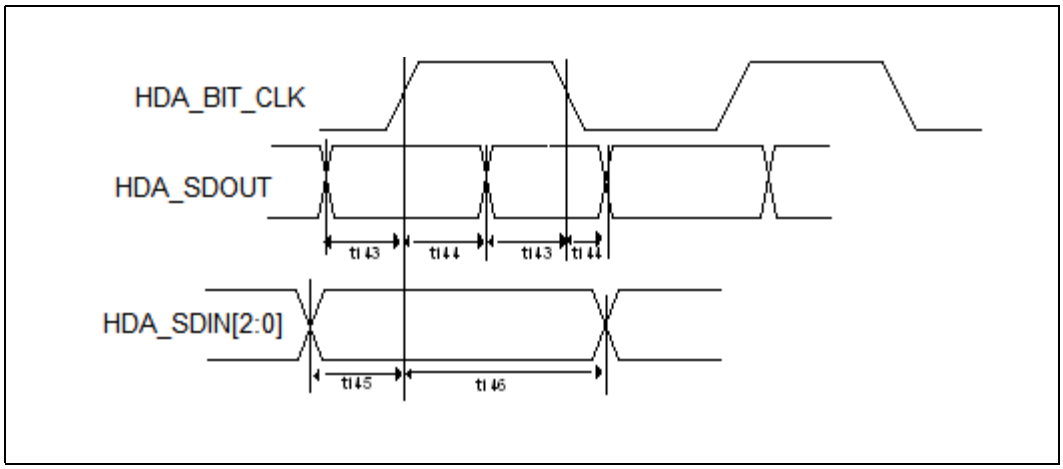
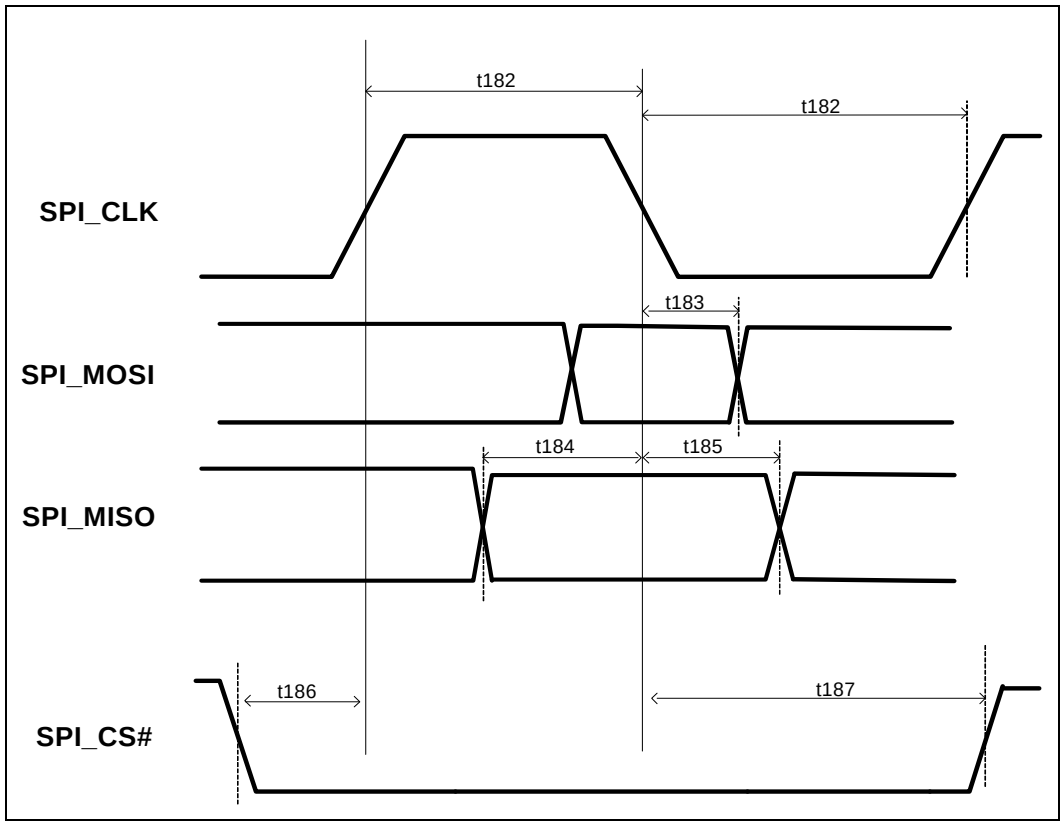


**Figure 8-45. SPI Timings**




§ §

# 9 Register and Memory Mapping

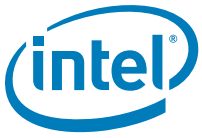
The Chipset contains registers that are located in the processor's I/O space and memory space and sets of PCI configuration registers that are located in PCI configuration space. This chapter describes the Chipset I/O and memory maps at the register-set level. Register access is also described. Register-level address maps and Individual register bit descriptions are provided in the following chapters. The following notations and definitions are used in the register/instruction description chapters.

| | |
|---|---|
| **RO** | Read Only. In some cases, If a register is read only, writes to this register location have no effect. However, in other cases, two separate registers are located at the same location where a read accesses one of the registers and a write accesses the other register. See the I/O and memory map tables for details. |
| **WO** | Write Only. In some cases, If a register is write only, reads to this register location have no effect. However, in other cases, two separate registers are located at the same location where a read accesses one of the registers and a write accesses the other register. See the I/O and memory map tables for details. |
| **R/W** | Read/Write. A register with this attribute can be read and written. |
| **R/WC** | Read/Write Clear. A register bit with this attribute can be read and written. However, a write of 1 clears (sets to 0) the corresponding bit and a write of 0 has no effect. |
| **R/WO** | Read/Write-Once. A register bit with this attribute can be written only once after power up. After the first write, the bit becomes read only. |
| **R/WLO** | Read/Write, Lock-Once. A register bit with this attribute can be written to the non-locked value multiple times, but to the locked value only once. After the locked value has been written, the bit becomes read only. |
| **Default** | When Chipset is reset, it sets its registers to predetermined default states. The default state represents the minimum functionality feature set required to successfully bring up the system. Hence, it does not represent the optimal system configuration. It is the responsibility of the system initialization software to determine configuration, operating parameters, and optional system features that are applicable, and to program the Chipset registers accordingly. |
| **Bold** | Register bits that are highlighted in bold text indicate that the bit is implemented in the Chipset. Register bits that are not implemented or are hardwired will remain in plain text. |

All bit(s) or bit-fields must be correctly dealt with by software. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit locations are preserved. Any Chipset configuration register or I/O or memory mapped location not explicitly indicated in this document must be considered reserved.

## 9.1 PCI Devices and Functions

The Chipset incorporates a variety of PCI functions as shown in Table 9-106. These functions are divided into six logical devices (B0:D30, B0:D31, B0:D29, B0:D28, B0:D27 and B1:D8). D30 contains the DMI interface-to-PCI bridge and the AC'97 Audio and Modem controller. D31 contains the PCI-to-LPC bridge, SATA controller, and the



SMBus controller. D29 contains the four USB UHCI controllers and one USB EHCI controller. D27 contains the Intel High Definition Audio controller. B1:D8 is the integrated LAN controller.

***Note:*** From a software perspective, the integrated LAN controller resides on the Chipset's external PCI bus. This is typically Bus 1, but may be assigned a different number depending on system configuration.

If for some reason, the particular system platform does not want to support any one of the Device Functions, with the exception of D30:F0, they can individually be disabled. The integrated LAN controller will be disabled if no Platform LAN Connect component is detected (See Chapter 5.3 - Volume 1). When a function is disabled, it does not appear at all to the software. A disabled function will not respond to any register reads or writes, insuring that these devices appear hidden to software.

**Table 9-106.PCI Devices and Functions**

| Bus:Device:Function[1] | Function Description |
|---|---|
| Bus 0:Device 30:Function 0 | PCI-to-PCI Bridge |
| Bus 0:Device 31:Function 0 | LPC Controller[1] |
| Bus 0:Device 31:Function 2 | SATA Controller |
| Bus 0:Device 31:Function 3 | SMBus Controller |
| Bus 0:Device 29:Function 0 | USB UHCI Controller #1 |
| Bus 0:Device 29:Function 1 | USB UHCI Controller #2 |
| Bus 0:Device 29:Function 2 | USB UHCI Controller #3 |
| Bus 0:Device 29:Function 3 | USB UHCI Controller #4 |
| Bus 0:Device 29:Function 7 | USB 2.0 EHCI Controller |
| Bus 0:Device 28:Function 0 | PCI Express* Port 1 |
| Bus 0:Device 28:Function 1 | PCI Express Port 2 |
| Bus 0:Device 28:Function 2 | PCI Express Port 3 |
| Bus 0:Device 28:Function 3 | PCI Express Port 4 |
| Bus 0:Device 27:Function 0 | Intel HD Audio Controller |
| Bus n:Device 8:Function 0 | LAN Controller |

**NOTES:**
1. The LPC controller contains registers that control LPC, Power Management, System Management, GPIO, processor Interface, RTC, Interrupts, Timers, DMA.

## 9.2 PCI Configuration Map

Each PCI function on the Chipset has a set of PCI configuration registers. The register address map tables for these register sets are included at the beginning of the chapter for the particular function. Configuration Space registers are accessed through configuration cycles on the PCI bus by the Host bridge using configuration mechanism #1 detailed in the *PCI Local Bus Specification, Revision 2.3*.

Some of the PCI registers contain reserved bits. Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back. Note the software does not need to perform read, merge, write operation for the configuration address register.

In addition to reserved bits within a register, the configuration space contains reserved locations. Software should not write to reserved PCI configuration locations in the device-specific region (above address offset 3Fh).

## 9.3 I/O Map

The I/O map is divided into Fixed and Variable address ranges. Fixed ranges cannot be moved, but in some cases can be disabled. Variable ranges can be moved and can also be disabled.

### 9.3.1 Fixed I/O Address Ranges

Table 9-107 shows the Fixed I/O decode ranges from the processor perspective. Note that for each I/O range, there may be separate behavior for reads and writes. DMI (Direct Media Interface) cycles that go to target ranges that are marked as “Reserved” will not be decoded by the Chipset, and will be passed to PCI unless the Subtractive Decode Policy bit is set (D31:F0:Offset 42h, bit 0). If a PCI master targets one of the fixed I/O target ranges, it will be positively decoded by the Chipset in medium speed.

Address ranges that are not listed or marked “Reserved” are **not** decoded by the Chipset (unless assigned to one of the variable ranges).

**Table 9-107.Fixed I/O Ranges Decoded by Chipset (Sheet 1 of 3)**

| I/O Address | Read Target | Write Target | Internal Unit |
|---|---|---|---|
| 00h–08h | DMA Controller | DMA Controller | DMA |
| 09h–0Eh | RESERVED | DMA Controller | DMA |
| 0Fh | DMA Controller | DMA Controller | DMA |
| 10h–18h | DMA Controller | DMA Controller | DMA |
| 19h–1Eh | RESERVED | DMA Controller | DMA |
| 1Fh | DMA Controller | DMA Controller | DMA |
| 20h–21h | Interrupt Controller | Interrupt Controller | Interrupt |
| 24h–25h | Interrupt Controller | Interrupt Controller | Interrupt |
| 28h–29h | Interrupt Controller | Interrupt Controller | Interrupt |
| 2Ch–2Dh | Interrupt Controller | Interrupt Controller | Interrupt |
| 2E–2F | LPC SIO | LPC SIO | Forwarded to LPC |

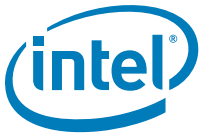


**Table 9-107.Fixed I/O Ranges Decoded by Chipset (Sheet 2 of 3)**

| I/O Address | Read Target | Write Target | Internal Unit |
|---|---|---|---|
| 30h–31h | Interrupt Controller | Interrupt Controller | Interrupt |
| 34h–35h | Interrupt Controller | Interrupt Controller | Interrupt |
| 38h–39h | Interrupt Controller | Interrupt Controller | Interrupt |
| 3Ch–3Dh | Interrupt Controller | Interrupt Controller | Interrupt |
| 40h–42h | Timer/Counter | Timer/Counter | PIT (8254) |
| 43h | RESERVED | Timer/Counter | PIT |
| 4E–4F | LPC SIO | LPC SIO | Forwarded to LPC |
| 50h–52h | Timer/Counter | Timer/Counter | PIT |
| 53h | RESERVED | Timer/Counter | PIT |
| 60h | Microcontroller | Microcontroller | Forwarded to LPC |
| 61h | NMI Controller | NMI Controller | Processor I/F |
| 62h | Microcontroller | Microcontroller | Forwarded to LPC |
| 64h | Microcontroller | Microcontroller | Forwarded to LPC |
| 66h | Microcontroller | Microcontroller | Forwarded to LPC |
| 70h | RESERVED | NMI and RTC Controller | RTC |
| 71h | RTC Controller | RTC Controller | RTC |
| 72h | RTC Controller | NMI and RTC Controller | RTC |
| 73h | RTC Controller | RTC Controller | RTC |
| 74h | RTC Controller | NMI and RTC Controller | RTC |
| 75h | RTC Controller | RTC Controller | RTC |
| 76h | RTC Controller | NMI and RTC Controller | RTC |
| 77h | RTC Controller | RTC Controller | RTC |
| 80h | DMA Controller, or LPC, or PCI | DMA Controller and LPC or PCI | DMA |
| 81h–83h | DMA Controller | DMA Controller | DMA |
| 84h–86h | DMA Controller | DMA Controller and LPC or PCI | DMA |
| 87h | DMA Controller | DMA Controller | DMA |
| 88h | DMA Controller | DMA Controller and LPC or PCI | DMA |
| 89h–8Bh | DMA Controller | DMA Controller | DMA |
| 8Ch–8Eh | DMA Controller | DMA Controller and LPC or PCI | DMA |
| 08Fh | DMA Controller | DMA Controller | DMA |
| 90h–91h | DMA Controller | DMA Controller | DMA |
| 92h | Reset Generator | Reset Generator | Processor I/F |
| 93h–9Fh | DMA Controller | DMA Controller | DMA |

**Table 9-107.Fixed I/O Ranges Decoded by Chipset (Sheet 3 of 3)**

| I/O Address | Read Target | Write Target | Internal Unit |
|---|---|---|---|
| A0h–A1h | Interrupt Controller | Interrupt Controller | Interrupt |
| A4h–A5h | Interrupt Controller | Interrupt Controller | Interrupt |
| A8h–A9h | Interrupt Controller | Interrupt Controller | Interrupt |
| ACh–ADh | Interrupt Controller | Interrupt Controller | Interrupt |
| B0h–B1h | Interrupt Controller | Interrupt Controller | Interrupt |
| B2h–B3h | Power Management | Power Management | Power Management |
| B4h–B5h | Interrupt Controller | Interrupt Controller | Interrupt |
| B8h–B9h | Interrupt Controller | Interrupt Controller | Interrupt |
| BCh–BDh | Interrupt Controller | Interrupt Controller | Interrupt |
| C0h–D1h | DMA Controller | DMA Controller | DMA |
| D2h–DDh | RESERVED | DMA Controller | DMA |
| DEh–DFh | DMA Controller | DMA Controller | DMA |
| F0h | FERR#/IGNNE# / Interrupt Controller | FERR#/IGNNE# / Interrupt Controller | Processor I/F |
| 170h–177h | IDE Controller, SATA Controller, or PCI | SATA Controller, or PCI | Forwarded to IDE or SATA |
| 1F0h–1F7h | IDE Controller, SATA Controller, or PCI [1] | SATA Controller, or PCI | Forwarded to IDE or SATA |
| 376h | IDE Controller, SATA Controller, or PCI | SATA Controller, or PCI | Forwarded to IDE or SATA |
| 3F6h | IDE Controller, SATA Controller, or PCI [1] | SATA Controller, or PCI | Forwarded IDE or SATA |
| 4D0h–4D1h | Interrupt Controller | Interrupt Controller | Interrupt |
| CF9h | Reset Generator | Reset Generator | Processor I/F |

**NOTES:**
1. A read to this address will subtractively go to PCI, where it will master abort.

## 9.3.2 Variable I/O Decode Ranges

Table 9-108 shows the Variable I/O Decode Ranges. They are set using Base Address Registers (BARs) or other configuration bits in the various PCI 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. Unpredictable results if the configuration software allows conflicts to occur. The Chipset does not perform any checks for conflicts.

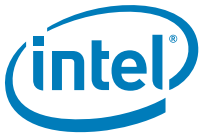


**Table 9-108.Variable I/O Decode Ranges**

| Range Name | Mappable | Size (Bytes) | Target |
|---|---|---|---|
| ACPI | Anywhere in 64 KB I/O Space | 64 | Power Management |
| IDE Bus Master | Anywhere in 64 KB I/O Space | 16 | IDE Unit |
| Native IDE Command | Anywhere in 64 KB I/O Space | 8 | IDE Unit |
| Native IDE Control | Anywhere in 64 KB I/O Space | 4 | IDE Unit |
| USB UHCI Controller #1 | Anywhere in 64 KB I/O Space | 32 | USB Unit 1 |
| USB UHCI Controller #2 | Anywhere in 64 KB I/O Space | 32 | USB Unit 2 |
| USB UHCI Controller #3 | Anywhere in 64 KB I/O Space | 32 | USB Unit 3 |
| USB UHCI Controller #4 | Anywhere in 64 KB I/O Space | 32 | USB Unit 4 |
| SMBus | Anywhere in 64 KB I/O Space | 32 | SMB Unit |
| TCO | 96 Bytes above ACPI Base | 32 | TCO Unit |
| GPIO | Anywhere in 64 KB I/O Space | 64 | GPIO Unit |
| Parallel Port | 3 Ranges in 64 KB I/O Space | 8 | LPC Peripheral |
| Serial Port 1 | 8 Ranges in 64 KB I/O Space | 8 | LPC Peripheral |
| Serial Port 2 | 8 Ranges in 64 KB I/O Space | 8 | LPC Peripheral |
| Floppy Disk Controller | 2 Ranges in 64 KB I/O Space | 8 | LPC Peripheral |
| LAN | Anywhere in 64 KB I/O Space | 64 | LAN Unit |
| LPC Generic 1 | Anywhere in 64 KB I/O Space | 4 to 256 | LPC Peripheral |
| LPC Generic 2[1] | Anywhere in 64 KB I/O Space | 4 to 256 | LPC Peripheral |
| LPC Generic 3 | Anywhere in 64 KB I/O Space | 4 to 256 | LPC Peripheral |
| LPC Generic 4 | Anywhere in 64 KB I/O Space | 4 to 256 | LPC Peripheral |
| I/O Trapping Ranges | Anywhere in 64 KB I/O Space | 1 to 256 | Trap on Backbone |

**NOTE:**
1. Decode range size determined by D31:F0:ADh:bits 5:4

## 9.4 Memory Map

Table 9-109 shows (from the processor perspective) the memory ranges that the Chipset decodes. Cycles that arrive from DMI that are not directed to any of the internal memory targets that decode directly from DMI will be driven out on PCI unless the Subtractive Decode Policy bit is set (D31:F0:Offset 42h, bit 0). The Chipset may then claim the cycle for the internal LAN controller.

PCI cycles generated by external PCI masters will be positively decoded unless they fall in the PCI-to-PCI bridge memory forwarding ranges (those addresses are reserved for PCI peer-to-peer traffic). If the cycle is not in the internal LAN controller's range, it will be forwarded up to DMI. Software must not attempt locks to the Chipset's memory-mapped I/O ranges for EHCI and HPET. If attempted, the lock is not honored which means potential deadlock conditions may occur.

### Table 9-109.Memory Decode Ranges from Processor Perspective (Sheet 1 of 2)

| Memory Range | Target | Dependency/Comments |
|---|---|---|
| 0000 0000h–000D FFFFh<br>0010 0000h–TOM<br>(Top of Memory) | Main Memory | TOM registers in Host controller |
| 000E 0000h–000E FFFFh | Firmware Hub | Bit 6 in Firmware Hub Decode Enable register is set |
| 000F 0000h–000F FFFFh | Firmware Hub | Bit 7 in Firmware Hub Decode Enable register is set |
| FEC0 0000h–FEC0 0100h | I/O APIC inside Chipset | |
| FED4 0000h–FED4 0FFFh | TPM on LPC | |
| FFC0 0000h–FFC7 FFFFh<br>FF80 0000h–FF87 FFFFh | Firmware Hub (or PCI)[1] | Bit 8 in Firmware Hub Decode Enable register is set |
| FFC8 0000h–FFCF FFFFh<br>FF88 0000h–FF8F FFFFh | Firmware Hub (or PCI)[1] | Bit 9 in Firmware Hub Decode Enable register is set |
| FFD0 0000h–FFD7 FFFFh<br>FF90 0000h–FF97 FFFFh | Firmware Hub (or PCI)[1] | Bit 10 in Firmware Hub Decode Enable register is set |
| FFD8 0000h–FFDF FFFFh<br>FF98 0000h–FF9F FFFFh | Firmware Hub (or PCI)[1] | Bit 11 in Firmware Hub Decode Enable register is set |
| FFE0 000h–FFE7 FFFFh<br>FFA0 0000h–FFA7 FFFFh | Firmware Hub (or PCI)[1] | Bit 12 in Firmware Hub Decode Enable register is set |
| FFE8 0000h–FFEF FFFFh<br>FFA8 0000h–FFAF FFFFh | Firmware Hub (or PCI)[1] | Bit 13 in Firmware Hub Decode Enable register is set |
| FFF0 0000h–FFF7 FFFFh<br>FFB0 0000h–FFB7 FFFFh | Firmware Hub (or PCI)[1] | Bit 14 in Firmware Hub Decode Enable register is set |
| FFF8 0000h–FFFF FFFFh<br>FFB8 0000h–FFBF FFFFh | Firmware Hub (or PCI)[1] | Always enabled.<br>The top two, 64 KB blocks of this range can be swapped, as described in Section 7.4.1. |
| FF70 0000h–FF7F FFFFh<br>FF30 0000h–FF3F FFFFh | Firmware Hub (or PCI)[1] | Bit 3 in Firmware Hub Decode Enable register is set |
| FF60 0000h–FF6F FFFFh<br>FF20 0000h–FF2F FFFFh | Firmware Hub (or PCI)[1] | Bit 2 in Firmware Hub Decode Enable register is set |
| FF50 0000h–FF5F FFFFh<br>FF10 0000h–FF1F FFFFh | Firmware Hub (or PCI)[1] | Bit 1 in Firmware Hub Decode Enable register is set |
| FF40 0000h–FF4F FFFFh<br>FF00 0000h–FF0F FFFFh | Firmware Hub (or PCI)[1] | Bit 0 in Firmware Hub Decode Enable register is set |
| 4 KB anywhere in 4-GB range | Integrated LAN Controller[2] | Enable via BAR in Device 29:Function 0 (Integrated LAN Controller) |
| 1 KB anywhere in 4-GB range | USB EHCI Controller [2] | Enable via standard PCI mechanism (Device 29, Function 7) |

**Table 9-109.Memory Decode Ranges from Processor Perspective (Sheet 2 of 2)**

| Memory Range | Target | Dependency/Comments |
|---|---|---|
| 512 B anywhere in 64-bit addressing space | Intel HD Audio Host Controller | Enable via standard PCI mechanism (Device 30, Function 1) |
| FED0 X000h–FED0 X3FFh | High Precision Event Timers [3] | BIOS determines the “fixed” location which is one of four, 1-KB ranges where X (in the first column) is 0h, 1h, 2h, or 3h. |
| All other | PCI | None |

**NOTES:**
1. PCI is the target when the Boot BIOS Destination selection bit is low (Chipset Config Registers:Offset 3401:bit 3). When PCI selected, the Firmware Hub Decode Enable bits have no effect.
2. Only LAN cycles can be seen on PCI.
3. Software must not attempt locks to memory mapped I/O ranges for USB EHCI or High Precision Event Timers. If attempted, the lock is not honored, which means potential deadlock conditions may occur.

## 9.4.1 Boot-Block Update Scheme

The Chipset supports a “top-block swap” mode that has the Chipset swap the top block in the Firmware Hub (the boot block) with another location. This allows for safe update of the Boot Block (even if a power failure occurs). When the “TOP_SWAP” Enable bit is set, the Chipset will invert A16 for cycles targeting Firmware Hub space. When this bit is 0, the Chipset will not invert A16. 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_SWAP bit. This will invert A16 for cycles going to the Firmware Hub. processor access to FFFF_0000h through FFFF_FFFFh will be directed to FFFE_0000h through FFFE_FFFFh in the Firmware Hub, and processor accesses to FFFE_0000h through FFFE_FFFF will be directed to FFFF_0000h through FFFF_FFFFh.
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_SWAP bit
8. Software sets the Top_Swap Lock-Down bit

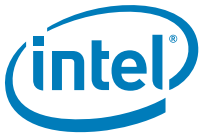
If a power failure occurs at any point after step 3, the system will be able to boot from the copy of the boot block that is stored in the block below the top. This is because the TOP_SWAP bit is backed in the RTC well.

***Note:*** The top-block swap mode may be forced by an external strapping option (See Section 2.22.1 - Volume 1). When top-block swap mode is forced in this manner, the TOP_SWAP bit cannot be cleared by software. A re-boot with the strap removed will be required to exit a forced top-block swap mode.

***Note:*** Top-block swap mode only affects accesses to the Firmware Hub space, not feature space.

***Note:*** The top-block swap mode has no effect on accesses below FFFE_0000h.

§



# 10 Chipset Configuration Registers

This chapter describes all registers and base functionality that is related to chipset configuration and not a specific interface (e.g., LPC, PCI, or PCI Express*). It contains the root complex register block that describes the behavior of the upstream internal link.

This block is mapped into memory space, using register RCBA of the PCI-to-LPC bridge. Accesses in this space must be limited to 32-(DWord) bit quantities. Burst accesses are not allowed.

## 10.1 Chipset Configuration Registers (Memory Space)

***Note:*** Address locations that are not shown should be treated as Reserved (see Section 9.2 for details).

**Table 10-110.Chipset Configuration Register Memory Map (Memory Space) (Sheet 1 of 3)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 0000–0003h | VCH | Virtual Channel Capability Header | 10010002h | RO |
| 0004–0007h | VCAP1 | Virtual Channel Capability #1 | 00000801h | RO |
| 0008–000Bh | VCAP2 | Virtual Channel Capability #2 | 00000001h | RO |
| 000C–000Dh | PVC | Port Virtual Channel Control | 0000h | R/W, RO |
| 000E–000Fh | PVS | Port Virtual Channel Status | 0000h | RO |
| 0010–0013h | V0CAP | Virtual Channel 0 Resource Capability | 00000001h | RO |
| 0014–0017h | V0CTL | Virtual Channel 0 Resource Control | 800000FFh | R/W, RO |
| 001A–001Bh | V0STS | Virtual Channel 0 Resource Status | 0000h | RO |
| 001C–001Fh | V1CAP | Virtual Channel 1 Resource Capability | 30008010h | R/WO, RO |
| 0020–0023h | V1CTL | Virtual Channel 1 Resource Control | 00000000h | R/W, RO |
| 0026–0027h | V1STS | Virtual Channel 1 Resource Status | 0000h | RO |
| 0100–0103h | RCTCL | Root Complex Topology Capability List | 1A010005h | RO |
| 0104–0107h | ESD | Element Self Description | 00000602h | R/WO, RO |
| 0110–0113h | ULD | Upstream Link Descriptor | 00000001h | R/WO, RO |
| 0118–011Fh | ULBA | Upstream Link Base Address | 0000000000000000h | R/WO |
| 0120–0123h | RP1D | Root Port 1 Descriptor | 01xx0002h | R/WO, RO |
| 0128–012Fh | RP1BA | Root Port 1 Base Address | 00000000000E0000h | RO |

**Table 10-110.Chipset Configuration Register Memory Map (Memory Space) (Sheet 2 of 3)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 0130–0133h | RP2D | Root Port 2 Descriptor | 02xx0002h | R/WO, RO |
| 0138–013Fh | RP2BA | Root Port 2 Base Address | 00000000000E1000h | RO |
| 0140–0143h | RP3D | Root Port 3 Descriptor | 03xx0002h | R/WO, RO |
| 0148–014Fh | RP3BA | Root Port 3 Base Address | 00000000000E2000h | RO |
| 0150–0153h | RP4D | Root Port 4 Descriptor | 04xx0002h | R/WO, RO |
| 0158–015Fh | RP4BA | Root Port 4 Base Address | 00000000000E3000h | RO |
| 0160–0163h | HDD | Intel HD Audio Descriptor | 05xx0002h | R/WO, RO |
| 0168–016Fh | HDBA | Intel HD Audio Base Address | 00000000000D8000h | RO |
| 01A0–01A3h | ILCL | Internal Link Capability List | 00010006h | RO |
| 01A4–01A7h | LCAP | Link Capabilities | 00012441h | RO, R/WO |
| 01A8–01A9h | LCTL | Link Control | 0000h | R/W |
| 01AA–01ABh | LSTS | Link Status | 0041h | RO |
| 0224–0227h | RPC | Root Port Configuration | 0000000xh | R/W, RO |
| 0238–023Bh | RPFN | Root Port Function Number for PCI Express Root Ports | 00543210h | R/WO, RO |
| 1E00–1E03h | TRSR | Trap Status Register | 00h | R/WC, RO |
| 1E10–1E17h | TRCR | Trapped Cycle Register | 0000000000000000h | RO |
| 1E18-1E1Fh | TWDR | Trapped Write Data Register | 0000000000000000h | RO |
| 1E80-1E87h | IOTR0 | I/O Trap Register 0 | 0000000000000000h | R/W, RO |
| 1E88-1E8Fh | IOTR1 | I/O Trap Register 1 | 0000000000000000h | R/W, RO |
| 1E90-1E97h | IOTR2 | I/O Trap Register 2 | 0000000000000000h | R/W, RO |
| 1E98-1E9Fh | IOTR3 | I/O Trap Register 3 | 0000000000000000h | R/W, RO |
| 3000–3001h | TCTL | TCO Control | 00h | R/W |
| 3100–3103h | D31IP | Device 31 Interrupt Pin | 00042210h | R/W, RO |
| 3104–3107h | D30IP | Device 30 Interrupt Pin | 00002100h | R/W, RO |
| 3108–310Bh | D29IP | Device 29 Interrupt Pin | 10004321h | R/W |
| 310C–310Fh | D28IP | Device 28 Interrupt Pin | 00004321h | R/W |
| 3110–3113h | D27IP | Device 27 Interrupt Pin | 00000001h | R/W |
| 3140–3141h | D31IR | Device 31 Interrupt Route | 3210h | R/W |
| 3142–3143h | D30IR | Device 30 Interrupt Route | 3210h | R/W |
| 3144–3145h | D29IR | Device 29 Interrupt Route | 3210h | R/W |
| 3146–3147h | D28IR | Device 28 Interrupt Route | 3210h | R/W |
| 3148–3149h | D27IR | Device 27 Interrupt Route | 3210h | R/W |
| 31FF–31FFh | OIC | Other Interrupt Control | 00h | R/W |
| 3400–3403h | RC | RTC Configuration | 00000000h | R/W, R/WLO |
| 3404–3407h | HPTC | High Precision Timer Configuration | 00000000h | R/W |
| 3410–3413h | GCS | General Control and Status | 0000000xh | R/W, R/WLO |

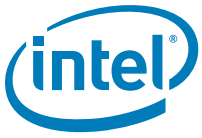


**Table 10-110.Chipset Configuration Register Memory Map (Memory Space) (Sheet 3 of 3)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 3414–3414h | BUC | Backed Up Control | 0000001xb | R/W |
| 3418–341Bh | FD | Function Disable | See bit description | R/W, RO |
| 341C–341Fh | CG | Clock Gating (Netbook Only) | 00000000h | R/W, RO |

## 10.1.1 VCH—Virtual Channel Capability Header Register

Offset Address: 0000–0003h Attribute: RO
Default Value: 10010002h Size: 32-bit

| Bit | Description |
|---|---|
| 31:20 | **Next Capability Offset (NCO)** — RO.This field indicates the next item in the list. |
| 19:16 | **Capability Version (CV)** — RO. This field indicates support as a version 1 capability structure. |
| 15:0 | **Capability ID (CID)** — RO. This field indicates this is the Virtual Channel capability item. |

## 10.1.2 VCAP1—Virtual Channel Capability #1 Register

Offset Address: 0004–0007h Attribute: RO
Default Value: 00000801h Size: 32-bit

| Bit | Description |
|---|---|
| 31:12 | **Reserved** |
| 11:10 | **Port Arbitration Table Entry Size (PATS)** — RO. This field indicates the size of the port arbitration table is 4 bits (to allow up to 8 ports). |
| 9:8 | **Reference Clock (RC)** — RO. Fixed at 100 ns. |
| 7 | **Reserved** |
| 6:4 | **Low Priority Extended VC Count (LPEVC)** — RO. This field indicates that there are no additional VCs of low priority with extended capabilities. |
| 3 | **Reserved** |
| 2:0 | **Extended VC Count (EVC)** — RO. This field indicates that there is one additional VC (VC1) that exists with extended capabilities. |

### 10.1.3 VCAP2—Virtual Channel Capability #2 Register

Offset Address: 0008–000Bh Attribute: RO
Default Value: 00000001h Size: 32-bit

| Bit | Description |
|---|---|
| 31:24 | **VC Arbitration Table Offset (ATO)** — RO. This field indicates that no table is present for VC arbitration since it is fixed. |
| 23:8 | **Reserved** |
| 7:0 | **VC Arbitration Capability (AC)** — RO. This field indicates that the VC arbitration is fixed in the root complex. VC1 is highest priority and VC0 is lowest priority. |

### 10.1.4 PVC—Port Virtual Channel Control Register

Offset Address: 000C–000Dh Attribute: R/W, RO
Default Value: 0000h Size: 16-bit

| Bit | Description |
|---|---|
| 15:04 | **Reserved** |
| 3:1 | **VC Arbitration Select (AS)** — RO. This field indicates which VC should be programmed in the VC arbitration table. The root complex takes no action on the setting of this field since there is no arbitration table. |
| 0 | **Load VC Arbitration Table (LAT)** — RO. This bit indicates that the table programmed should be loaded into the VC arbitration table. This bit is defined as read/write with always returning 0 on reads. |

### 10.1.5 PVS—Port Virtual Channel Status Register

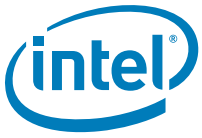
Offset Address: 000E–000Fh Attribute: RO
Default Value: 0000h Size: 16-bit

| Bit | Description |
|---|---|
| 15:01 | **Reserved** |
| 0 | **VC Arbitration Table Status (VAS)** — RO. This bit indicates the coherency status of the VC Arbitration table when it is being updated. This field is always 0 in the root complex since there is no VC arbitration table. |

### 10.1.6 V0CAP—Virtual Channel 0 Resource Capability Register

Offset Address: 0010–0013h Attribute: RO
Default Value: 00000001h Size: 32-bit

| Bit | Description |
|---|---|
| 31:24 | **Port Arbitration Table Offset (AT)** — RO. This VC implements no port arbitration table since the arbitration is fixed. |
| 23 | **Reserved** |
| 22:16 | **Maximum Time Slots (MTS)** — RO. This VC implements fixed arbitration, and therefore this field is not used. |



| Bit | Description |
|---|---|
| 15 | **Reject Snoop Transactions (RTS)** — RO. This VC must be able to take snoopable transactions. |
| 14 | **Advanced Packet Switching (APS)** — RO. This VC is capable of all transactions, not just advanced packet switching transactions. |
| 13:8 | **Reserved** |
| 7:0 | **Port Arbitration Capability (PAC)** — RO. This field indicates that this VC uses fixed port arbitration. |

## 10.1.7 V0CTL—Virtual Channel 0 Resource Control Register

Offset Address: 0014–0017h Attribute: R/W, RO
Default Value: 800000FFh Size: 32-bit

| Bit | Description |
|---|---|
| 31 | **Virtual Channel Enable (EN)** — RO. Always set to 1. VC0 is always enabled and cannot be disabled. |
| 30:27 | **Reserved** |
| 26:24 | **Virtual Channel Identifier (ID)** — RO. This field indicates the ID to use for this virtual channel. |
| 23:20 | **Reserved** |
| 19:17 | **Port Arbitration Select (PAS)** — R/W. This field indicates which port table is being programmed. The root complex takes no action on this setting since the arbitration is fixed and there is no arbitration table. |
| 16 | **Load Port Arbitration Table (LAT)** — RO. The root complex does not implement an arbitration table for this virtual channel. |
| 15:8 | **Reserved** |
| 7:1 | **Transaction Class / Virtual Channel Map (TVM)** — R/W. This field indicates which transaction classes are mapped to this virtual channel. When a bit is set, this transaction class is mapped to the virtual channel. |
| 0 | **Reserved** |

## 10.1.8 V0STS—Virtual Channel 0 Resource Status Register

Offset Address: 001A–001Bh Attribute: RO
Default Value: 0000h Size: 16-bit

| Bit | Description |
|---|---|
| 15:02 | **Reserved** |
| 1 | **VC Negotiation Pending (NP)** — RO. When set, this bit indicates the virtual channel is still being negotiated with ingress ports. |
| 0 | **Port Arbitration Tables Status (ATS)** — RO. There is no port arbitration table for this VC, so this bit is **reserved** at 0. |

### 10.1.9 V1CAP—Virtual Channel 1 Resource Capability Register

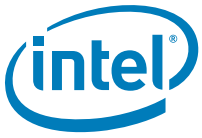
Offset Address: 001C–001Fh Attribute: R/WO, RO
Default Value: 30008010h Size: 32-bit

| Bit | Description |
|---|---|
| 31:24 | **Port Arbitration Table Offset (AT)** — RO. This field indicates the location of the port arbitration table in the root complex. A value of 3h indicates the table is at offset 30h. |
| 23 | **Reserved** |
| 22:16 | **Maximum Time Slots (MTS)** — R/WO. This value is updated by platform BIOS based upon the determination of the number of time slots available in the platform. |
| 15 | **Reject Snoop Transactions (RTS)** — RO. All snoopable transactions on VC1 are rejected. This VC is for isochronous transfers only. |
| 14 | **Advanced Packet Switching (APS)** — RO. This VC is capable of all transactions, not just advanced packet switching transactions. |
| 13:8 | **Reserved** |
| 7:0 | **Port Arbitration Capability (PAC)** — RO. This field indicates the port arbitration capability is time-based WRR of 128 phases. |

### 10.1.10 V1CTL—Virtual Channel 1 Resource Control Register

Offset Address: 0020–0023h Attribute: R/W, RO
Default Value: 00000000h Size: 32-bit

| Bit | Description |
|---|---|
| 31 | **Virtual Channel Enable (EN)** — R/W.<br>0 = Disables the VC.<br>1 = Enables the VC. |
| 30:27 | **Reserved** |
| 26:24 | Virtual Channel Identifier (ID) — R/W. This field indicates the ID to use for this virtual channel. |
| 23:20 | **Reserved** |
| 19:17 | **Port Arbitration Select (PAS)** — R/W. This field indicates which port table is being programmed. The only permissible value of this field is 4h for the time-based WRR entries. |
| 16 | **Load Port Arbitration Table (LAT)** — RO/W. When set, the port arbitration table loaded is based upon the PAS field in this register. This bit always returns 0 when read. |
| 15:8 | **Reserved** |
| 7:1 | **Transaction Class / Virtual Channel Map (TVM)** — R/W. This field indicates which transaction classes are mapped to this virtual channel. When a bit is set, this transaction class is mapped to the virtual channel. |
| 0 | **Reserved** |



### 10.1.11 V1STS—Virtual Channel 1 Resource Status Register

Offset Address: 0026–0027h Attribute: RO
Default Value: 0000h Size: 16-bit

| Bit | Description |
|---|---|
| 15:02 | **Reserved** |
| 1 | **VC Negotiation Pending (NP)** — RO.<br>0 = Virtual channel is Not being negotiated with ingress ports.<br>1 = The virtual channel is still being negotiated with ingress ports. |
| 0 | **Port Arbitration Tables Status (ATS)** — RO. This field indicates the coherency status of the port arbitration table. This bit is set when LAT (offset 000Ch:bit 0) is written with value 1 and PAS (offset 0014h:bits19:17) has value of 4h. This bit is cleared after the table has been updated. |

### 10.1.12 RCTCL—Root Complex Topology Capabilities List Register

Offset Address: 0100–0103h Attribute: RO
Default Value: 1A010005h Size: 32-bit

| Bit | Description |
|---|---|
| 31:20 | **Next Capability (NEXT)** — RO. This field indicates the next item in the list. |
| 19:16 | **Capability Version (CV)** — RO. This field indicates the version of the capability structure. |
| 15:0 | **Capability ID (CID)** — RO. This field indicates this is a PCI Express* link capability section of an RCRB. |

### 10.1.13 ESD—Element Self Description Register

Offset Address: 0104–0107h Attribute: R/WO, RO
Default Value: 00000602h Size: 32-bit

| Bit | Description |
|---|---|
| 31:24 | **Port Number (PN)** — RO. A value of 0 to indicate the egress port for the Chipset. |
| 23:16 | **Component ID (CID) — R/WO**. This field indicates the component ID assigned to this element by software. This is written once by platform BIOS and is locked until a platform reset. |
| 15:8 | **Number of Link Entries (NLE)** — RO. This field indicates that one link entry (corresponding to DMI), 6 root port entries (for the downstream ports), and the Intel HD Audio device are described by this RCRB. |
| 7:4 | **Reserved** |
| 3:0 | **Element Type (ET)** — RO. This field indicates that the element type is a root complex internal link. |

### 10.1.14 ULD—Upstream Link Descriptor Register

Offset Address: 0110–0113h Attribute: R/WO, RO
Default Value: 00000001h Size: 32-bit

| Bit | Description |
|---|---|
| 31:24 | **Target Port Number (PN)** — R/WO. This field is programmed by platform BIOS to match the port number of the (G)MCH/CPU RCRB that is attached to this RCRB. |
| 23:16 | **Target Component ID (TCID)** — R/WO. This field is programmed by platform BIOS to match the component ID of the (G)MCH/CPU RCRB that is attached to this RCRB. |
| 15:2 | **Reserved** |
| 1 | **Link Type (LT)** — RO. This bit indicates that the link points to the (G)MCH/CPU RCRB. |
| 0 | **Link Valid (LV)** — RO. This bit indicates that the link entry is valid. |

### 10.1.15 ULBA—Upstream Link Base Address Register

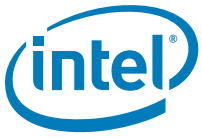
Offset Address: 0118–011Fh Attribute: R/WO
Default Value: 0000000000000000h Size: 64-bit

| Bit | Description |
|---|---|
| 63:32 | **Base Address Upper (BAU)** — R/WO. This field is programmed by platform BIOS to match the upper 32-bits of base address of the (G)MCH/CPU RCRB that is attached to this RCRB. |
| 31:0 | **Base Address Lower (BAL)** — R/WO. This field is programmed by platform BIOS to match the lower 32-bits of base address of the (G)MCH/CPU RCRB that is attached to this RCRB. |

### 10.1.16 RP1D—Root Port 1 Descriptor Register

Offset Address: 0120–0123h Attribute: R/WO, RO
Default Value: 01xx0002h Size: 32-bit

| Bit | Description |
|---|---|
| 31:24 | **Target Port Number (PN)** — RO. This field indicates that the target port number is 1h (root port #1). |
| 23:16 | **Target Component ID (TCID)** — R/WO. This field returns the value of the ESD.CID (offset 0104h, bits 23:16) field programmed by platform BIOS, since the root port is in the same component as the RCRB. |
| 15:2 | **Reserved** |
| 1 | **Link Type (LT)** — RO. This bit indicates that the link points to a root port. |
| 0 | **Link Valid (LV)** — RO. When FD.PE1D (offset 3418h, bit 16) is set, this link is not valid (returns 0). When FD.PE1D is cleared, this link is valid (returns 1). |



## 10.1.17 RP1BA—Root Port 1 Base Address Register

Offset Address: 0128–012Fh Attribute: RO
Default Value: 00000000000E0000h Size: 64-bit

| Bit | Description |
|---|---|
| 63:32 | **Reserved** |
| 31:28 | **Reserved** |
| 27:20 | **Bus Number (BN)** — RO. This field indicates the root port is on bus #0. |
| 19:15 | **Device Number (DN)** — RO. This field indicates the root port is on device #28. |
| 14:12 | **Function Number (FN)** — RO. This field indicates the root port is on function #0. |
| 11:0 | **Reserved** |

## 10.1.18 RP2D—Root Port 2 Descriptor Register

Offset Address: 0130–0133h Attribute: R/WO, RO
Default Value: 02xx0002h Size: 32-bit

| Bit | Description |
|---|---|
| 31:24 | **Target Port Number (PN)** — RO. This field indicates the target port number is 2h (root port #2). |
| 23:16 | **Target Component ID (TCID)** — R/WO. This field returns the value of the ESD.CID (offset 0104h, bits 23:16) field programmed by platform BIOS, since the root port is in the same component as the RCRB. |
| 15:2 | **Reserved** |
| 1 | **Link Type (LT)** — RO. This bit indicates that the link points to a root port. |
| 0 | **Link Valid (LV)** — RO. When RPC.PC (offset 0224h, bits 1:0) is ‘01’, ‘10’, or ‘11’, or FD.PE2D (offset 3418h, bit 17) is set, the link for this root port is not valid (return 0). When RPC.PC is ‘00’ and FD.PE2D is cleared, the link for this root port is valid (return 1). |

## 10.1.19 RP2BA—Root Port 2 Base Address Register

Offset Address: 0138–013Fh Attribute: RO
Default Value: 00000000000E1000h Size: 64-bit

| Bit | Description |
|---|---|
| 63:32 | **Reserved** |
| 31:28 | **Reserved** |
| 27:20 | **Bus Number (BN)** — RO. This field indicates the root port is on bus #0. |
| 19:15 | **Device Number (DN)** — RO. This field indicates the root port is on device #28. |
| 14:12 | **Function Number (FN)** — RO. This field indicates the root port is on function #1. |
| 11:0 | **Reserved** |

## 10.1.20 RP3D—Root Port 3 Descriptor Register

Offset Address: 0140–0143h Attribute: R/WO, RO
Default Value: 03xx0002h Size: 32-bit

| Bit | Description |
|---|---|
| 31:24 | **Target Port Number (PN)** — RO. This field indicates the target port number is 3h (root port #3). |
| 23:16 | **Target Component ID (TCID)** — R/WO. This field returns the value of the ESD.CID (offset 0104h, bits 23:16) field programmed by platform BIOS, since the root port is in the same component as the RCRB. |
| 15:2 | **Reserved** |
| 1 | **Link Type (LT)** — RO. This bit indicates that the link points to a root port. |
| 0 | **Link Valid (LV)** — RO. When RPC.PC (offset 0224h, bits 1:0) is '11', or FD.PE3D (offset 3418h, bit 18) is set, the link for this root port is not valid (return 0). When RPC.PC is '00', '01', or "10', and FD.PE3D is cleared, the link for this root port is valid (return 1). |

## 10.1.21 RP3BA—Root Port 3 Base Address Register

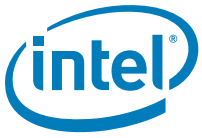
Offset Address: 0148–014Fh Attribute: RO
Default Value: 00000000000E2000h Size: 64-bit

| Bit | Description |
|---|---|
| 63:32 | **Reserved** |
| 31:28 | **Reserved** |
| 27:20 | **Bus Number (BN)** — RO. This field indicates the root port is on bus #0. |
| 19:15 | **Device Number (DN)** — RO. This field indicates the root port is on device #28. |
| 14:12 | **Function Number (FN)** — RO. This field indicates the root port is on function #2. |
| 11:0 | **Reserved** |

## 10.1.22 RP4D—Root Port 4 Descriptor Register

Offset Address: 0150–0153h Attribute: R/WO, RO
Default Value: 04xx0002h Size: 32-bit

| Bit | Description |
|---|---|
| 31:24 | **Target Port Number (PN)** — RO. This field indicates the target port number is 4h (root port #4). |
| 23:16 | **Target Component ID (TCID)** — R/WO. This field returns the value of the ESD.CID (offset 0104h, bits 23:16) field programmed by platform BIOS, since the root port is in the same component as the RCRB. |
| 15:2 | **Reserved** |
| 1 | **Link Type (LT)** — RO. This bit indicates that the link points to a root port. |
| 0 | **Link Valid (LV)** — RO. When RPC.PC (offset 0224h, bits 1:0) is '10' or '11', or FD.PE4D (offset 3418h, bit 19) is set, the link for this root port is not valid (return 0). When RPC.PC is '00' or '01' and FD.PE4D is cleared, the link for this root port is valid (return 1). |



### 10.1.23 RP4BA—Root Port 4 Base Address Register

Offset Address: 0158–015Fh Attribute: RO
Default Value: 00000000000E3000h Size: 64-bit

| Bit | Description |
|---|---|
| 63:32 | **Reserved** |
| 31:28 | **Reserved** |
| 27:20 | **Bus Number (BN)** — RO. This field indicates the root port is on bus #0. |
| 19:15 | **Device Number (DN)** — RO. This field indicates the root port is on device #28. |
| 14:12 | **Function Number (FN)** — RO. This field indicates the root port is on function #3. |
| 11:0 | **Reserved** |

### 10.1.24 HDD—Intel HD Audio Descriptor Register

Offset Address: 0160–0163h Attribute: R/WO, RO
Default Value: 15xx0002h Size: 32-bit

| Bit | Description |
|---|---|
| 31:24 | T**arget Port Number (PN)** — RO. This field indicates the target port number is 15h (Intel HD Audio). |
| 23:16 | **Target Component ID (TCID)** — R/WO. This field returns the value of the ESD.CID (offset 0104h, bits 23:16) field programmed by platform BIOS, since the root port is in the same component as the RCRB. |
| 15:2 | **Reserved** |
| 1 | **Link Type (LT)** — RO. This bit indicates that the link points to a root port. |
| 0 | **Link Valid (LV)** — RO. When FD.ZD (offset 3418h, bit 4) is set, the link to Intel High Definition Audio is not valid (return 0). When FD.ZD is cleared, the link to Intel High Definition Audio is valid (return 1). |

### 10.1.25 HDBA—Intel HD Audio Base Address Register

Offset Address: 0168–016Fh Attribute: RO
Default Value: 00000000000D8000h Size: 64-bit

| Bit | Description |
|---|---|
| 63:32 | **Reserved** |
| 31:28 | **Reserved** |
| 27:20 | **Bus Number (BN)** — RO. This field indicates the root port is on bus #0. |
| 19:15 | **Device Number (DN)** — RO. This field indicates the root port is on device #27. |
| 14:12 | **Function Number (FN)** — RO. This field indicates the root port is on function #0. |
| 11:0 | **Reserved** |

### 10.1.26 ILCL—Internal Link Capabilities List Register

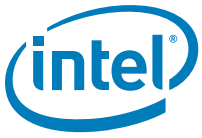
Offset Address: 01A0–01A3h Attribute: RO
Default Value: 00010006h Size: 32-bit

| Bit | Description |
|---|---|
| 31:20 | **Next Capability Offset (NEXT)** — RO. Indicates this is the last item in the list. |
| 19:16 | **Capability Version (CV)** — RO. This field indicates the version of the capability structure. |
| 15:0 | **Capability ID (CID)** — RO. This field indicates this is capability for DMI. |

### 10.1.27 LCAP—Link Capabilities Register

Offset Address: 01A4–01A7h Attribute: RO/ R/WO
Default Value: 00012441h Size: 32-bit

| Bit | Description |
|---|---|
| 31:18 | **Reserved** |
| 17:15 Nettop Only | **L1 Exit Latency (EL1)** — L1 not supported on DMI. |
| 17:15 Netbook Only | **L1 Exit Latency (EL1)** — RO. This field is set to 010b to indicate an exit latency of 2 us to 4 us. |
| 14:12 | **L0s Exit Latency (EL0)** — R/WO. This field indicates that exit latency is 128 ns to less than 256 ns. |
| 11:10 Nettop Only | **Active State Link PM Support (ASPM)** — R/WO. This field indicates that L0s is supported on DMI. |
| 11:10 Netbook Only | **Active State Link PM Support (ASPM)** — R/WO. This field indicates the level of active state power management on DMI.<br>00 = Neither L0s nor L1s are supported<br>01 = L0s Entry supported on DMI<br>10 = L1 Entry supported on DMI<br>11 = Both L0s and L1 supported on DMI |
| 9:4 | **Maximum Link Width (MLW)** — Indicates the maximum link width is 4 ports. |
| 3:0 | **Maximum Link Speed (MLS)** — Indicates the link speed is 2.5 Gb/s. |



## 10.1.28 LCTL—Link Control Register

Offset Address: 01A8–01A9h Attribute: R/W
Default Value: 0000h Size: 16-bit

| Bit | Description |
|---|---|
| 15:8 | **Reserved** |
| 7 | **Extended Synch (ES)** — R/W. When set, this bit forces extended transmission of FTS ordered sets when exiting L0s prior to entering L0 and extra sequences (Netbook Only) at exit from L1 prior to entering L0. |
| 6:2 | **Reserved** |
| 1:0 Nettop Only | **Active State Link PM Control (APMC)** — R/W. This field indicates whether DMI should enter L0s.<br>00 = Disabled<br>01 = L0s entry enabled<br>10 = **Reserved**<br>11 = **Reserved** |
| 1:0 Netbook Only | **Active State Link PM Control (APMC)** — R/W. This field indicates whether DMI should enter L0s or L1 or both.<br>00 = Disabled<br>01 = L0s entry enabled<br>10 = L1 Entry enabled<br>11 = L0s and L1 Entry enabled |

## 10.1.29 LSTS—Link Status Register

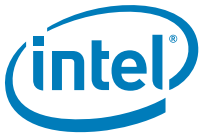
Offset Address: 01AA–01ABh Attribute: RO
Default Value: 0041h Size: 16-bit

| Bit | Description |
|---|---|
| 15:10 | **Reserved** |
| 9:4 | **Negotiated Link Width (NLW)** — RO. Negotiated link width is x4 (000100b). Netbook only: The Chipset may also indicate x2 (000010b), depending on CPU configuration. |
| 3:0 | **Link Speed (LS)** — RO. Link is 2.5 Gb/s. |

## 10.1.30 RPC—Root Port Configuration Register

Offset Address: 0224–0227h Attribute: R/W, RO
Default Value: 0000000yh (y = 00xxb) Size: 32-bit

| Bit | Description |
|---|---|
| 31:8 | **Reserved** |
| 7 | **High Priority Port Enable (HPE)** — R/W.<br>0 = The high priority path is not enabled.<br>1 = The port selected by the HPP field in this register is enabled for high priority. It will be arbitrated above all other VC0 (including integrated VC0) devices. |
| 6:4 | **High Priority Port (HPP)** — R/W. This field controls which port is enabled for high priority when the HPE bit in this register is set.<br>111 = **Reserved**<br>110 = **Reserved**<br>101 = **Reserved**<br>100 = **Reserved**<br>100 = Port 4<br>010 = Port 3<br>001 = Port 2<br>000 = Port 1 |
| 3:2 | **Reserved** |
| 1:0 | **Port Configuration (PC)** — RO. This field controls how the PCI bridges are organized in various modes of operation. For the following mappings, if a port is not shown, it is considered a x1 port with no connection.<br>These bits represent the strap values of ACZ_SDOUT (bit 1) and ACZ_SYNC (bit 0) when TP3 is not pulled low at the rising edge of PWROK.<br>11 = 1 x4, Port 1 (x4)<br>10 = **Reserved**<br>01 = **Reserved**<br>00 = 4 x1s, Port 1 (x1), Port 2 (x1), Port 3 (x1), Port 4 (x1)<br>These bits live in the resume well and are only reset by RSMRST#. |



## 10.1.31 RPFN—Root Port Function Number for PCI Express Root Ports

Offset Address:0238–023BhAttribute:R/WO, RO
Default Value: 00543210h Size: 32-bit

For the PCI Express root ports, the assignment of a function number to a root port is not fixed. BIOS may re-assign the function numbers on a port by port basis. This capability will allow BIOS to disable/hide any root port and have still have functions 0 thru N-1 where N is the total number of enabled root ports.

Port numbers will remain fixed to a physical root port.

The existing root port Function Disable registers operate on physical ports (not functions).

Port Configuration (1x4, 4x1, etc.) is not affected by the logical function number assignment and is associated with physical ports.

| Bit | Description |
|---|---|
| 31:15 | **Reserved** |
| 14:12 | **Root Port 4 Function Number (RP4FN)** — R/WO. These bits set the function number for PCI Express Root Port 4. This root port function number must be a unique value from the other root port function numbers. |
| 11 | **Reserved** |
| 10:8 | **Root Port 3 Function Number (RP3FN)** — R/WO. These bits set the function number for PCI Express Root Port 3. This root port function number must be a unique value from the other root port function numbers. |
| 7 | **Reserved** |
| 6:4 | **Root Port 2 Function Number (RP2FN)** — R/WO. These bits set the function number for PCI Express Root Port 2. This root port function number must be a unique value from the other root port function numbers. |
| 3 | **Reserved** |
| 2:0 | **Root Port 1 Function Number (RP1FN)** — R/WO. These bits set the function number for PCI Express Root Port 1. This root port function number must be a unique value from the other root port function numbers. |

### 10.1.32 TRSR—Trap Status Register

Offset Address: 1E00–1E03h Attribute: R/WC, RO
Default Value: 00000000h Size: 32-bit

| Bit | Description |
|---|---|
| 31:4 | **Reserved** |
| 3:0 | **Cycle Trap SMI# Status (CTSS)** — R/WC. These bits are set by hardware when the corresponding Cycle Trap register is enabled and a matching cycle is received (and trapped). These bits are OR'ed together to create a single status bit in the Power Management register space.<br>Note that the SMI# and trapping must be enabled in order to set these bits.<br>These bits are set before the completion is generated for the trapped cycle, thereby ensuring that the processor can enter the SMI# handler when the instruction completes. Each status bit is cleared by writing a 1 to the corresponding bit location in this register. |

### 10.1.33 TRCR—Trapped Cycle Register

Offset Address: 1E10–1E17h Attribute: RO
Default Value: 0000000000000000h Size: 64-bit

This register saves information about the I/O Cycle that was trapped and generated the SMI# for software to read.

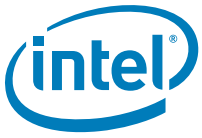
| Bit | Description |
|---|---|
| 63:25 | **Reserved** |
| 24 | **Read/Write# (RWI)** — RO.<br>0 = Trapped cycle was a write cycle.<br>1 = Trapped cycle was a read cycle. |
| 23:20 | **Reserved** |
| 19:16 | **Active-high Byte Enables (AHBE)** — RO. This is the DWord-aligned byte enables associated with the trapped cycle. A 1 in any bit location indicates that the corresponding byte is enabled in the cycle. |
| 15:2 | **Trapped I/O Address (TIOA)** — RO. This is the DWord-aligned address of the trapped cycle. |
| 1:0 | **Reserved** |

### 10.1.34 TWDR—Trapped Write Data Register

Offset Address: 1E18–1E1Fh Attribute: RO
Default Value: 0000000000000000h Size: 64-bit

This register saves the data from I/O write cycles that are trapped for software to read.

| Bit | Description |
|---|---|
| 63:32 | **Reserved** |
| 31:0 | **Trapped I/O Data (TIOD)** — RO. DWord of I/O write data. This field is undefined after trapping a read cycle. |



## 10.1.35 IOTRn — I/O Trap Register (0-3)

| | | | |
|---|---|---|---|
| Offset Address: | 1E80–1E87h Register 0<br>1E88–1E8Fh Register 1<br>1E90–1E97h Register 2<br>1E98–1E9Fh Register 3 | Attribute: | R/W, RO |
| Default Value: | 0000000000000000h | Size: | 64-bit |

These registers are used to specify the set of I/O cycles to be trapped and to enable this functionality.

| Bit | Description |
|---|---|
| 63:50 | **Reserved** |
| 49 | **Read/Write Mask (RWM)** — R/W.<br>0 = The cycle must match the type specified in bit 48.<br>1 = Trapping logic will operate on both read and write cycles. |
| 48 | **Read/Write# (RWIO)** — R/W.<br>0 = Write<br>1 = Read<br>**NOTE:** The value in this field does not matter if bit 49 is set. |
| 47:40 | **Reserved** |
| 39:36 | **Byte Enable Mask (BEM)** — R/W. A 1 in any bit position indicates that any value in the corresponding byte enable bit in a received cycle will be treated as a match. The corresponding bit in the Byte Enables field, below, is ignored. |
| 35:32 | **Byte Enables (TBE)** — R/W. Active-high DWord-aligned byte enables. |
| 31:24 | **Reserved** |
| 23:18 | **Address[7:2] Mask (ADMA)** — R/W. A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for traps on address ranges up to 256 bytes in size. |
| 17:16 | **Reserved** |
| 15:2 | **I/O Address[15:2] (IOAD)** — R/W. DWord-aligned address |
| 1 | **Reserved** |
| 0 | **Trap and SMI# Enable (TRSE)** — R/W.<br>0 = Trapping and SMI# logic disabled.<br>1 = The trapping logic specified in this register is enabled. |

### 10.1.36 TCTL—TCO Configuration Register

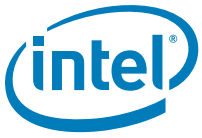
Offset Address: 3000–3000h Attribute: R/W
Default Value: 00h Size: 8-bit

| Bit | Description |
|---|---|
| 7 | **TCO IRQ Enable (IE)** — R/W.<br>0 = TCO IRQ is disabled.<br>1 = TCO IRQ is enabled, as selected by the TCO_IRQ_SEL field. |
| 6:3 | **Reserved** |
| 2:0 | **TCO IRQ Select (IS)** — R/W. Specifies on which IRQ the TCO will internally appear. If not using the APIC, the TCO interrupt must be routed to IRQ9-11, and that interrupt is not sharable with the SERIRQ stream, but is shareable with other PCI interrupts. If using the APIC, the TCO interrupt can also be mapped to IRQ20-23, and can be shared with other interrupt.<br>000 = IRQ 9<br>001 = IRQ 10<br>010 = IRQ 11<br>011 = **Reserved**<br>100 = IRQ 20 (only if APIC enabled)<br>101 = IRQ 21 (only if APIC enabled)<br>110 = IRQ 22 (only if APIC enabled)<br>111 = IRQ 23 (only if APIC enabled)<br><br>**NOTE:** When setting the these bits, the IE bit should be cleared to prevent glitching.<br>**NOTE:** When the interrupt is mapped to APIC interrupts 9, 10 or 11, the APIC should be programmed for active-high reception. When the interrupt is mapped to APIC interrupts 20 through 23, the APIC should be programmed for active-low reception. |

### 10.1.37 D31IP—Device 31 Interrupt Pin Register

Offset Address: 3100–3103h Attribute: R/W, RO
Default Value: 00042210h Size: 32-bit

| Bit | Description |
|---|---|
| 31:16 | **Reserved** |
| 15:12 | **SM Bus Pin (SMIP)** — R/W. This field indicates which pin the SMBus controller drives as its interrupt.<br>0h = No interrupt<br>1h = INTA#<br>2h = INTB# (Default)<br>3h = INTC#<br>4h = INTD#<br>5h–Fh = **Reserved** |



| Bit | Description |
|---|---|
| 11:8 | **SATA Pin (SIP)** — R/W. This field indicates which pin the SATA controller drives as its interrupt.<br>0h = No interrupt<br>1h = INTA#<br>2h = INTB# (Default)<br>3h = INTC#<br>4h = INTD#<br>5h–Fh = **Reserved** |
| 7:4 | **Reserved** |
| 3:0 | **PCI Bridge Pin (PCIP)** — RO. Currently, the PCI bridge does not generate an interrupt, so this field is read-only and 0. |

## 10.1.38 D30IP—Device 30 Interrupt Pin Register

Offset Address: 3104–3107h Attribute: R/W, RO
Default Value: 00002100h Size: 32-bit

| Bit | Description |
|---|---|
| 31:4 | **Reserved** |
| 3:0 | **LPC Bridge Pin (LIP)** — RO. Currently, the LPC bridge does not generate an interrupt, so this field is read-only and 0. |

## 10.1.39 D29IP—Device 29 Interrupt Pin Register

Offset Address: 3108–310Bh Attribute: R/W
Default Value: 10004321h Size: 32-bit

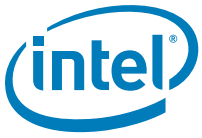
| Bit | Description |
|---|---|
| 31:28 | **EHCI Pin (EIP)** — R/W. This field indicates which pin the EHCI controller drives as its interrupt.<br>0h = No interrupt<br>1h = INTA# (Default)<br>2h = INTB#<br>3h = INTC#<br>4h = INTD#<br>5h–Fh = **Reserved** |
| 27:16 | **Reserved** |
| 15:12 | **UHCI #3 Pin (U3P)** — R/W. This field indicates which pin the UHCI controller #3 (ports 6 and 7) drives as its interrupt.<br>0h = No interrupt<br>1h = INTA#<br>2h = INTB#<br>3h = INTC#<br>4h = INTD# (Default)<br>5h–Fh = **Reserved** |

| Bit | Description |
|---|---|
| 11:8 | **UHCI #2 Pin (U2P)** — R/W. This field indicates which pin the UHCI controller #2 (ports 4 and 5) drives as its interrupt.<br>0h = No interrupt<br>1h = INTA#<br>2h = INTB#<br>3h = INTC# (Default)<br>4h = INTD#<br>5h–Fh = **Reserved** |
| 7:4 | **UHCI #1 Pin (U1P)** — R/W. This field indicates which pin the UHCI controller #1 (ports 2 and 3) drives as its interrupt.<br>0h = No interrupt<br>1h = INTA#<br>2h = INTB# (Default)<br>3h = INTC#<br>4h = INTD#<br>5h–Fh = **Reserved** |
| 3:0 | **UHCI #0 Pin (U0P)** — R/W. This field indicates which pin the UHCI controller #0 (ports 0 and 1) drives as its interrupt.<br>0h = No interrupt<br>1h = INTA# (Default)<br>2h = INTB#<br>3h = INTC#<br>4h = INTD#<br>5h–Fh = **Reserved** |

## 10.1.40 D28IP—Device 28 Interrupt Pin Register

Offset Address: 310C–310Fh Attribute: R/W
Default Value: 00214321h Size: 32-bit

| Bit | Description |
|---|---|
| 31:16 | **Reserved** |
| 15:12 | **PCI Express #4 Pin (P4IP)** — R/W. This field indicates which pin the PCI Express* port #4 drives as its interrupt.<br>0h = No interrupt<br>1h = INTA#<br>2h = INTB#<br>3h = INTC#<br>4h = INTD# (Default)<br>5h–Fh = **Reserved** |
| 11:8 | **PCI Express #3 Pin (P3IP)** — R/W. This field indicates which pin the PCI Express port #3 drives as its interrupt.<br>0h = No interrupt<br>1h = INTA#<br>2h = INTB#<br>3h = INTC# (Default)<br>4h = INTD#<br>5h–Fh = **Reserved** |



| Bit | Description |
|---|---|
| 7:4 | **PCI Express #2 Pin (P2IP)** — R/W. This field indicates which pin the PCI Express port #2 drives as its interrupt.<br>0h = No interrupt<br>1h = INTA#<br>2h = INTB# (Default)<br>3h = INTC#<br>4h = INTD#<br>5h–Fh = **Reserved** |
| 3:0 | **PCI Express #1 Pin (P1IP)** — R/W.This field indicates which pin the PCI Express port #1 drives as its interrupt.<br>0h = No interrupt<br>1h = INTA# (Default)<br>2h = INTB#<br>3h = INTC#<br>4h = INTD#<br>5h–Fh = **Reserved** |

## 10.1.41 D27IP—Device 27 Interrupt Pin Register

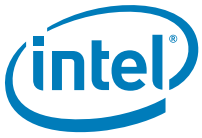
Offset Address: 3110–3113h Attribute: R/W
Default Value: 00000001h Size: 32-bit

| Bit | Description |
|---|---|
| 31:4 | **Reserved** |
| 3:0 | **Intel HD Audio Pin (ZIP)** — R/W. This field indicates which pin the Intel High Definition Audio controller drives as its interrupt.<br>0h = No interrupt<br>1h = INTA# (Default)<br>2h = INTB#<br>3h = INTC#<br>4h = INTD#<br>5h-Fh = **Reserved** |

## 10.1.42 D31IR—Device 31 Interrupt Route Register

Offset Address: 3140–3141h Attribute: R/W
Default Value: 3210h Size: 16-bit

| Bit | Description |
|---|---|
| 15 | **Reserved** |
| 14:12 | **Interrupt D Pin Route (IDR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTD# pin reported for device 31 functions.<br>0h = PIRQA#<br>1h = PIRQB#<br>2h = PIRQC#<br>3h = PIRQD# (Default)<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 11 | **Reserved** |
| 10:8 | **Interrupt C Pin Route (ICR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTC# pin reported for device 31 functions.<br>0h = PIRQA#<br>1h = PIRQB#<br>2h = PIRQC# (Default)<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 7 | **Reserved** |
| 6:4 | **Interrupt B Pin Route (IBR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTB# pin reported for device 31 functions.<br>0h = PIRQA#<br>1h = PIRQB# (Default)<br>2h = PIRQC#<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 3 | **Reserved** |



| Bit | Description |
|---|---|
| 2:0 | **Interrupt A Pin Route (IAR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTA# pin reported for device 31 functions.<br>0h = PIRQA# (Default)<br>1h = PIRQB#<br>2h = PIRQC#<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |

## 10.1.43 D30IR—Device 30 Interrupt Route Register

Offset Address: 3142–3143h Attribute: R/W
Default Value: 3210h Size: 16-bit

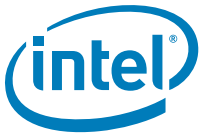
| Bit | Description |
|---|---|
| 15 | **Reserved** |
| 14:12 | **Interrupt D Pin Route (IDR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTD# pin reported for device 30 functions.<br>0h = PIRQA#<br>1h = PIRQB#<br>2h = PIRQC#<br>3h = PIRQD# (Default)<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 11 | **Reserved** |
| 10:8 | **Interrupt C Pin Route (ICR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTC# pin reported for device 30 functions.<br>0h = PIRQA#<br>1h = PIRQB#<br>2h = PIRQC# (Default)<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 7 | **Reserved** |

| Bit | Description |
|---|---|
| 6:4 | **Interrupt B Pin Route (IBR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTB# pin reported for device 30 functions.<br>0h = PIRQA#<br>1h = PIRQB# (Default)<br>2h = PIRQC#<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 3 | **Reserved** |
| 2:0 | **Interrupt A Pin Route (IAR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTA# pin reported for device 30 functions.<br>0h = PIRQA# (Default)<br>1h = PIRQB#<br>2h = PIRQC#<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |

### 10.1.44 D29IR—Device 29 Interrupt Route Register

Offset Address: 3144–3145h Attribute: R/W
Default Value: 3210h Size: 16-bit

| Bit | Description |
|---|---|
| 15 | **Reserved** |
| 14:12 | **Interrupt D Pin Route (IDR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTD# pin reported for device 29 functions.<br>0h = PIRQA#<br>1h = PIRQB#<br>2h = PIRQC#<br>3h = PIRQD# (Default)<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 11 | **Reserved** |

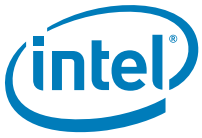


| Bit | Description |
|---|---|
| 10:8 | **Interrupt C Pin Route (ICR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTC# pin reported for device 29 functions.<br>0h = PIRQA#<br>1h = PIRQB#<br>2h = PIRQC# (Default)<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 7 | **Reserved** |
| 6:4 | **Interrupt B Pin Route (IBR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTB# pin reported for device 29 functions.<br>0h = PIRQA#<br>1h = PIRQB# (Default)<br>2h = PIRQC#<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 3 | **Reserved** |
| 2:0 | **Interrupt A Pin Route (IAR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTA# pin reported for device 29 functions.<br>0h = PIRQA# (Default)<br>1h = PIRQB#<br>2h = PIRQC#<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |

## 10.1.45 D28IR—Device 28 Interrupt Route Register

Offset Address: 3146–3147h Attribute: R/W
Default Value: 3210h Size: 16-bit

| Bit | Description |
|---|---|
| 15 | **Reserved** |
| 14:12 | **Interrupt D Pin Route (IDR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTD# pin reported for device 28 functions.<br>0h = PIRQA#<br>1h = PIRQB#<br>2h = PIRQC#<br>3h = PIRQD# (Default)<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 11 | **Reserved** |
| 10:8 | **Interrupt C Pin Route (ICR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTC# pin reported for device 28 functions.<br>0h = PIRQA#<br>1h = PIRQB#<br>2h = PIRQC# (Default)<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 7 | **Reserved** |
| 6:4 | **Interrupt B Pin Route (IBR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTB# pin reported for device 28 functions.<br>0h = PIRQA#<br>1h = PIRQB# (Default)<br>2h = PIRQC#<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 3 | **Reserved** |
| 2:0 | **Interrupt A Pin Route (IAR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTA# pin reported for device 28 functions.<br>0h = PIRQA# (Default)<br>1h = PIRQB#<br>2h = PIRQC#<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |



## 10.1.46 D27IR—Device 27 Interrupt Route Register

Offset Address: 3148–3149h Attribute: R/W
Default Value: 3210h Size: 16-bit

| Bit | Description |
|---|---|
| 15 | **Reserved** |
| 14:12 | **Interrupt D Pin Route (IDR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTD# pin reported for device 27 functions.<br>0h = PIRQA#<br>1h = PIRQB#<br>2h = PIRQC#<br>3h = PIRQD# (Default)<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 11 | **Reserved** |
| 10:8 | **Interrupt C Pin Route (ICR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTC# pin reported for device 27 functions.<br>0h = PIRQA#<br>1h = PIRQB#<br>2h = PIRQC# (Default)<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 7 | **Reserved** |
| 6:4 | **Interrupt B Pin Route (IBR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTB# pin reported for device 27 functions.<br>0h = PIRQA#<br>1h = PIRQB# (Default)<br>2h = PIRQC#<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |
| 3 | **Reserved** |
| 2:0 | **Interrupt A Pin Route (IAR)** — R/W. This field indicates which physical pin on the Chipset is connected to the INTA# pin reported for device 27 functions.<br>0h = PIRQA# (Default)<br>1h = PIRQB#<br>2h = PIRQC#<br>3h = PIRQD#<br>4h = PIRQE#<br>5h = PIRQF#<br>6h = PIRQG#<br>7h = PIRQH# |

### 10.1.47 OIC—Other Interrupt Control Register

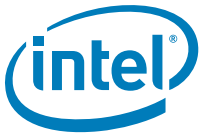
Offset Address: 31FF–31FFh Attribute: R/W
Default Value: 00h Size: 8-bit

| Bit | Description |
|---|---|
| 7:2 | **Reserved** |
| 1 | **Coprocessor Error Enable (CEN)** — R/W.<br>0 = FERR# will not generate IRQ13 nor IGNNE#.<br>1 = If FERR# is low, the Chipset generates IRQ13 internally and holds it until an I/O port F0h write. It will also drive IGNNE# active. |
| 0 | **APIC Enable (AEN)** — R/W.<br>0 = The internal IOxAPIC is disabled.<br>1 = Enables the internal IOxAPIC and its address decode. |

### 10.1.48 RC—RTC Configuration Register

Offset Address: 3400–3403h Attribute: R/W, R/WLO
Default Value: 00000000h Size: 32-bit

| Bit | Description |
|---|---|
| 31:5 | **Reserved** |
| 4 | **Upper 128 Byte Lock (UL)** — R/WLO.<br>0 = Bytes not locked.<br>1 = Bytes 38h–3Fh in the upper 128-byte bank of RTC RAM are locked and cannot be accessed. Writes will be dropped and reads will not return valid data. This bit is reset on system reset. |
| 3 | **Lower 128 Byte Lock (LL)** — R/WLO.<br>0 = Bytes not locked.<br>1 = Bytes 38h–3Fh in the lower 128-byte bank of RTC RAM are locked and cannot be accessed. Writes will be dropped and reads will not return valid data. Bit reset on system reset. |
| 2 | **Upper 128 Byte Enable (UE)** — R/W.<br>0 = Bytes locked.<br>1 = The upper 128-byte bank of RTC RAM can be accessed. |
| 1:0 | **Reserved** |



## 10.1.49 HPTC—High Precision Timer Configuration Register

Offset Address: 3404–3407h Attribute: R/W
Default Value: 00000000h Size: 32-bit

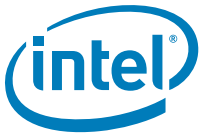
| Bit | Description |
|---|---|
| 31:8 | **Reserved** |
| 7 | **Address Enable (AE)** — R/W.<br>0 = Address disabled.<br>1 = The Chipset will decode the High Precision Timer memory address range selected by bits 1:0 below. |
| 6:2 | **Reserved** |
| 1:0 | **Address Select (AS)** — R/W. This field selects 1 of 4 possible memory address ranges for the High Precision Timer functionality. The encodings are:<br>00 = FED0_0000h – FED0_03FFh<br>01 = FED0_1000h – FED0_13FFh<br>10 = FED0_2000h – FED0_23FFh<br>11 = FED0_3000h – FED0_33FFh |

## 10.1.50 GCS—General Control and Status Register

Offset Address: 3410–3413h Attribute:R/W, R/WLO
Default Value: 00000yy0h (yy = xx0000x0b)Size: 32-bit

| Bit | Description |
|---|---|
| 31:12 | **Reserved** |
| 11:10 | **Boot BIOS Straps (BBS)**: This field determines the destination of accesses to the BIOS memory range. The default values for these bits represent the strap values of GNT5#/GPIO17 (bit 11) and GNT4#/GPIO48 (bit 10) (active-high logic levels) at the rising edge of PWROK.<br><br>**Bits 11:10** / **Description**<br>00b — **Reserved**<br>01b — SPI (supports shared flash with LAN)<br>10b — PCI<br>11b — LPC<br><br>When PCI is selected, the top 16 MB of memory below 4 GB (FF00_0000h to FFFF_FFFFh) is accepted by the primary side of the PCI-to-PCI bridge and forwarded to the PCI bus. This allows systems with corrupted or unprogrammed flash to boot from a PCI device. The PCI-to-PCI bridge Memory Space Enable bit does not need to be set (nor any other bits) for these cycles to go to PCI. Note that BIOS decode range bits and the other BIOS protection bits have no effect when PCI is selected.<br>When SPI or LPC is selected, the range that is decoded is further qualified by other configuration bits described in the respective sections.<br>The value in this field can be overwritten by software as long as the BIOS Interface Lock-Down (bit 0) is not set. |

| Bit | Description |
|---|---|
| 9 | **Server Error Reporting Mode (SERM)** — R/W.<br>0 = The Chipset is the final target of all errors. The (G)MCH/CPU sends a messages to the Chipset for the purpose of generating NMI.<br>1 = The (G)MCH/CPU is the final target of all errors from PCI Express* and DMI. In this mode, if the Chipset detects a fatal, non-fatal, or correctable error on DMI or its downstream ports, it sends a message to the (G)MCH/CPU. If the Chipset receives an ERR_* message from the downstream port, it sends that message to the (G)MCH/CPU. |
| 8:7 | **Reserved** |
| 6 (Netbook Only) | **FERR# MUX Enable (FME)** — R/W. This bit enables FERR# to be a processor break event indication.<br>0 = Disabled.<br>1 = The Chipset examines FERR# during a C2, C3, or C4 state as a break event.<br>See Chapter 5.14.5 - Volume 1 for a functional description. |
| 6 (Nettop Only) | **Reserved** |
| 5 | **No Reboot (NR)** — R/W. This bit is set when the “No Reboot” strap (SPKR pin on Chipset) is sampled high on PWROK. This bit may be set or cleared by software if the strap is sampled low but may not override the strap when it indicates “No Reboot”.<br>0 = System will reboot upon the second timeout of the TCO timer.<br>1 = The TCO timer will count down and generate the SMI# on the first timeout, but will not reboot on the second timeout. |
| 4 | **Alternate Access Mode Enable (AME)** — R/W.<br>0 = Disabled.<br>1 = Alternate access read only registers can be written, and write only registers can be read. 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 Chipset implements an alternate access mode. For a list of these registers see Section 5.14.10. |
| 3 | **Reserved**. |
| 2 | **Reserved Page Route (RPR)** — R/W. This bit determines where to send the **reserved** page registers. These addresses are sent to PCI or LPC for the purpose of generating POST codes. The I/O addresses modified by this field are: 80h, 84h, 85h, 86h, 88h, 8Ch, 8Dh, and 8Eh.<br>0 = Writes will be forwarded to LPC, shadowed within the Chipset, and reads will be returned from the internal shadow<br>1 = Writes will be forwarded to PCI, shadowed within the Chipset, and reads will be returned from the internal shadow.<br><br>**NOTE:** If some writes are completed to LPC/PCI to these I/O ranges, and then this bit is flipped such that writes will now go to the other interface, the reads will not return what was last written. Shadowing is performed on each interface.<br>The aliases for these registers, at 90h, 94h, 95h, 96h, 98h, 9Ch, 9Dh, and 9Eh, are always decoded to LPC. |
| 1 | **Reserved** |
| 0 | **BIOS Interface Lock-Down (BILD)** — R/WLO.<br>0 = Disabled.<br>1 = Prevents BUC.TS (offset 3414, bit 0) and GCS.BBS (offset 3410h, bits 11:10) from being changed. This bit can only be written from 0 to 1 once. |



## 10.1.51 BUC—Backed Up Control Register

Offset Address: 3414–3414h Attribute:R/W
Default Value: 0000001xb

All bits in this register are in the RTC well and only cleared by RTCRST#.

| Bit | Description |
|---|---|
| 7:3 | **Reserved** |
| 2 | **CPU BIST Enable (CBE)** — R/W. This bit is in the resume well and is reset by RSMRST#, but not PLTRST# nor CF9h writes.<br>0 = Disabled.<br>1 = The INIT# signals will be driven active when CPURST# is active. INIT# and INIT3_3V# will go inactive with the same timings as the other processor I/F signals (hold time after CPURST# inactive). |
| 1 | **Reserved** |
| 0 | **Top Swap (TS)** — R/W.<br>0 = Chipset will not invert A16.<br>1 = Chipset will invert A16 for cycles going to the BIOS space (but not the feature space) in the FWH.<br>If the Chipset is strapped for Top-Swap (STRAP0# is low at rising edge of PWROK), then this bit cannot be cleared by software. The strap jumper should be removed and the system rebooted. |

## 10.1.52 FD—Function Disable Register

Offset Address: 3418–341Bh Attribute: R/W, RO
Default Value: See bit description Size: 32-bit

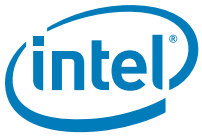
The UHCI functions must be disabled from highest function number to lowest. For example, if only three UHCIs are wanted, software must disable UHCI #4 (UD4 bit set). When disabling UHCIs, the EHCI Structural Parameters Registers must be updated with coherent information in “Number of Companion Controllers” and “N_Ports” fields.

When disabling a function, only the configuration space is disabled. Software must ensure that all functionality within a controller that is not desired (such as memory spaces, I/O spaces, and DMA engines) is disabled prior to disabling the function.

When a function is disabled, software must not attempt to re-enable it. A disabled function can only be re-enabled by a platform reset.

| Bit | Description |
|---|---|
| 31:20 | **Reserved** |
| 19 | **PCI Express 4 Disable (PE4D)** — R/W. When disabled, the link for this port is put into the “link down” state.<br>0 = PCI Express* port #4 is enabled. (Default)<br>1 = PCI Express port #4 is disabled. |
| 18 | **PCI Express 3 Disable (PE3D)** — R/W. When disabled, the link for this port is put into the link down state.<br>0 = PCI Express port #3 is enabled. (Default)<br>1 = PCI Express port #3 is disabled. |

| Bit | Description |
|---|---|
| 17 | **PCI Express 2 Disable (PE2D)** — R/W. When disabled, the link for this port is put into the link down state.<br>0 = PCI Express port #2 is enabled. (Default)<br>1 = PCI Express port #2 is disabled |
| 16 | **PCI Express 1 Disable (PE1D)** — R/W. When disabled, the link for this port is put into the link down state.<br>0 = PCI Express port #1 is enabled. (Default)<br>1 = PCI Express port #1 is disabled. |
| 15 | **EHCI Disable (EHCID)** — R/W.<br>0 = The EHCI is enabled. (Default)<br>1 = The EHCI is disabled. |
| 14 | **LPC Bridge Disable (LBD)** — R/W.<br>0 = The LPC bridge is enabled. (Default)<br>1 = The LPC bridge is disabled. Unlike the other disables in this register, the following additional spaces will no longer be decoded by the LPC bridge:<br>• Memory cycles below 16 MB (1000000h)<br>• I/O cycles below 64 KB (10000h)<br>• The Internal I/OxAPIC at FEC0_0000 to FECF_FFFF<br>Memory cycles in the LPC BIOS range below 4 GB will still be decoded when this bit is set, but the aliases at the top of 1 MB (the E and F segment) no longer will be decoded. |
| 13:12 | **Reserved** |
| 11 | **UHCI #4 Disable (U4D)** — R/W.<br>0 = The 4th UHCI (ports 6 and 7) is enabled. (Default)<br>1 = The 4th UHCI (ports 6 and 7) is disabled. |
| 10 | **UHCI #3 Disable (U3D)** — R/W.<br>0 = The 3rd UHCI (ports 4 and 5) is enabled. (Default)<br>1 = The 3rd UHCI (ports 4 and 5) is disabled. |
| 9 | **UHCI #2 Disable (U2D)** — R/W.<br>0 = The 2nd UHCI (ports 2 and 3) is enabled. (Default)<br>1 = The 2nd UHCI (ports 2 and 3) is disabled. |
| 8 | **UHCI #1 Disable (U1D)** — R/W.<br>0 = The 1st UHCI (ports 0 and 1) is enabled. (Default)<br>1 = The 1st UHCI (ports 0 and 1) is disabled. |
| 7 | **Hide Internal LAN (HIL)** — R/W.<br>0 = The LAN controller is enabled. (Default)<br>1 = The LAN controller is disabled and will not decode configuration cycles off of P |
| 6:5 | **Reserved** |
| 4 | **Intel HD Audio Disable (ZD)** — R/W.<br>0 = The Intel High Definition Audio controller is enabled. (Default)<br>1 = The Intel High Definition Audio controller is disabled and its PCI configuration space is not accessible. |
| 3 | **SM Bus Disable (SD)** — R/W.<br>0 = The SM Bus controller is enabled. (Default)<br>1 = The SM Bus controller is disabled. In Chipset and previous, this also disabled the I/O space. In the Chipset, it only disables the configuration space. |



| Bit | Description |
|---|---|
| 2 | **Serial ATA Disable (SAD)** — R/W. Default is 0.<br>0 = The SATA controller is enabled.<br>1 = The SATA controller is disabled. |
| 1:0 | **Reserved** |

## 10.1.53 CG—Clock Gating (Netbook only)

Offset Address: 341C–341Fh Attribute: R/W, RO
Default Value: 00000000h Size: 32-bit

| Bit | Description |
|---|---|
| 31 | **Legacy (LPC) Dynamic Clock Gate Enable** — R/W.<br>0 = Legacy Dynamic Clock Gating is Disabled<br>1 = Legacy Dynamic Clock Gating is Enabled |
| 30 | **Reserved** |
| 29:28 | **USB UHCI Dynamic Clock Gate Enable** — R/W.<br>0 = USB UHCI Dynamic Clock Gating is Disabled<br>1 = USB UHCI Dynamic Clock Gating is Enabled<br>**0 = Reserved**<br>**1 = Reserved** |
| 27:26 | **Reserved** |
| 25 | **SATA Port 1 Dynamic Clock Gate Enable** — R/W.<br>0 = SATA Port 1 Dynamic Clock Gating is Disabled<br>1 = SATA Port 1 Dynamic Clock Gating is Enabled |
| 24 | **SATA Port 0 Dynamic Clock Gate Enable** — R/W.<br>0 = SATA Port 0 Dynamic Clock Gating is Disabled<br>1 = SATA Port 0 Dynamic Clock Gating is Enabled |
| 23 | **Reserved** |
| 22 | **High Definition Audio Dynamic Clock Gate Enable** — R/W.<br>0 = HIgh Definition Audio Dynamic Clock Gating is Disabled<br>1 = HIgh Definition Audio Dynamic Clock Gating is Enabled |
| 21 | **High Definition Audio Static Clock Gate Enable** — R/W.<br>0 = HIgh Definition Audio Static Clock Gating is Disabled<br>1 = HIgh Definition Audio Static Clock Gating is Enabled |
| 20 | **USB EHCI Static Clock Gate Enable** — R/W.<br>0 = USB EHCI Static Clock Gating is Disabled<br>1 = USB EHCI Static Clock Gating is Enabled |
| 19 | **USB EHCI Dynamic Clock Gate Enable** — R/W.<br>0 = USB EHCI Dynamic Clock Gating is Disabled<br>1 = USB EHCI Dynamic Clock Gating is Enabled |
| 18:17 | **Reserved** |
| 16 | **PCI Dynamic Gate Enable** — R/W. Functionality **reserved**. BIOS must ensure bit is 0. |
| 15:4 | **Reserved** |

| Bit | Description |
|---|---|
| 3 | **DMI and PCI Express* RX Dynamic Clock Gate Enable** — R/W.<br>0 = DMI and PCI Express root port RX Dynamic Clock Gating is Disabled<br>1 = DMI and PCI Express root port RX Dynamic Clock Gating is Enabled |
| 2 | **PCI Express TX Dynamic Clock Gate Enable** — R/W.<br>0 = PCI Express root port TX Dynamic Clock Gating is Disabled<br>1 = PCI Express root port TX Dynamic Clock Gating is Enabled |
| 1 | **DMI TX Dynamic Clock Gate Enable** — R/W.<br>0 = DMI TX Dynamic Clock Gating is Disabled<br>1 = DMI TX Dynamic Clock Gating is Enabled |
| 0 | **PCI Express Root Port Static Clock Gate Enable** — R/W.<br>0 = PCI Express root port Static Clock Gating is Disabled<br>1 = PCI Express root port Static Clock Gating is Enabled |

§

# 11 LAN Controller Registers (B1:D8:F0)

The Chipset integrated LAN controller appears to reside at PCI Device 8, Function 0 on the secondary side of the Chipset's virtual PCI-to-PCI bridge. This is typically Bus 1, but may be assigned a different number depending upon system configuration. The LAN controller acts as both a master and a slave on the PCI bus. As a master, the LAN controller interacts with the system main memory to access data for transmission or deposit received data. As a slave, some of the LAN controller's control structures are accessed by the host processor to read or write information to the on-chip registers. The processor also provides the LAN controller with the necessary commands and pointers that allow it to process receive and transmit data.

## 11.1 PCI Configuration Registers (LAN Controller—B1:D8:F0)

***Note:*** Address locations that are not shown should be treated as **Reserved** (See Section 9.2 for details).

**Table 11-111.LAN Controller PCI Register Address Map (LAN Controller—B1:D8:F0) (Sheet 1 of 2)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00h–01h | VID | Vendor Identification | 8086h | RO |
| 02h–03h | DID | Device Identification | See register description. | RO |
| 04h–05h | PCICMD | PCI Command | 0000h | RO, R/W |
| 06h–07h | PCISTS | PCI Status | 0290h | RO, R/WC |
| 08h | RID | Revision Identification | See register description. | RO |
| 0Ah | SCC | Sub Class Code | 00h | RO |
| 0Bh | BCC | Base Class Code | 02 | RO |
| 0Ch | CLS | Cache Line Size | 00h | R/W |
| 0Dh | PMLT | Primary Master Latency Timer | 00h | R/W |
| 0Eh | HEADTYP | Header Type | 00h | RO |
| 10h–13h | CSR_MEM_BASE | CSR Memory–Mapped Base Address | 00000008h | R/W, RO |
| 14h–17h | CSR_IO_BASE | CSR I/O–Mapped Base Address | 00000001h | R/W, RO |
| 2Ch–2Dh | SVID | Subsystem Vendor Identification | 0000h | RO |
| 2Eh–2Fh | SID | Subsystem Identification | 0000h | RO |
| 34h | CAP_PTR | Capabilities Pointer | DCh | RO |
| 3Ch | INT_LN | Interrupt Line | 00h | R/W |
| 3Dh | INT_PN | Interrupt Pin | 01h | RO |

**Table 11-111.LAN Controller PCI Register Address Map (LAN Controller—B1:D8:F0) (Sheet 2 of 2)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 3Eh | MIN_GNT | Minimum Grant | 08h | RO |
| 3Fh | MAX_LAT | Maximum Latency | 38h | RO |
| DCh | CAP_ID | Capability ID | 01h | RO |
| DDh | NXT_PTR | Next Item Pointer | 00h | RO |
| DEh–DFh | PM_CAP | Power Management Capabilities | FE21h | RO |
| E0h–E1h | PMCSR | Power Management Control/ Status | 0000h | R/W, RO, R/WC |
| E3 | PCIDATA | PCI Power Management Data | 00h | RO |

## 11.1.1 VID—Vendor Identification Register (LAN Controller—B1:D8:F0)

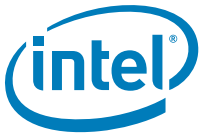
Offset Address: 00h–01h Attribute: RO
Default Value: 8086h Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Vendor ID** — RO. This is a 16-bit value assigned to Intel. |

## 11.1.2 DID—Device Identification Register (LAN Controller—B1:D8:F0)

Offset Address: 02h–03h Attribute: RO
Default Value: See bit description Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Device ID** — RO. This is a 16-bit value assigned to the chipset integrated LAN controller.<br><br>**NOTES:**<br>1. If the EEPROM is not present (or not properly programmed), reads to the Device ID return the default value referred to in the Intel® I/O Controller Hub 7 Family Specification Update.<br>2. If the EEPROM is present (and properly programmed) and if the value of word 23h is not 0000h or FFFFh, the Device ID is loaded from the EEPROM, word 23h after the hardware reset. (See Section 11.1.4 - SID, Subsystem ID of LAN controller for detail) |



### 11.1.3 PCICMD—PCI Command Register (LAN Controller—B1:D8:F0)

Offset Address: 04h–05h Attribute: RO, R/W
Default Value: 0000h Size: 16 bits

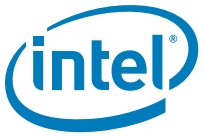
| Bit | Description |
|---|---|
| 15:11 | **Reserved** |
| 10 | **Interrupt Disable** — R/W.<br>0 = Enable.<br>1 = Disables LAN controller to assert its INTA signal. |
| 9 | **Fast Back to Back Enable (FBE)** — RO. Hardwired to 0. The integrated LAN controller will not run fast back-to-back PCI cycles. |
| 8 | **SERR# Enable (SERR_EN)** — R/W.<br>0 = Disable.<br>1 = Enable. Allow SERR# to be asserted. |
| 7 | Wait Cycle Control (WCC) — RO. Hardwired to 0. Not implemented. |
| 6 | **Parity Error Response (PER)** — R/W.<br>0 = The LAN controller will ignore PCI parity errors.<br>1 = The integrated LAN controller will take normal action when a PCI parity error is detected and will enable generation of parity on DMI. |
| 5 | VGA Palette Snoop (VPS) — RO. Hardwired to 0. Not Implemented. |
| 4 | **Memory Write and Invalidate Enable (MWIE)** — R/W.<br>0 = Disable. The LAN controller will not use the Memory Write and Invalidate command.<br>1 = Enable. |
| 3 | Special Cycle Enable (SCE) — RO. Hardwired to 0. The LAN controller ignores special cycles. |
| 2 | **Bus Master Enable (BME)** — R/W.<br>0 = Disable.<br>1 = Enable. The Chipset's integrated LAN controller may function as a PCI bus master. |
| 1 | **Memory Space Enable (MSE)** — R/W.<br>0 = Disable.<br>1 = Enable. The Chipset's integrated LAN controller will respond to the memory space accesses. |
| 0 | **I/O Space Enable (IOSE)** — R/W.<br>0 = Disable.<br>1 = Enable. The Chipset's integrated LAN controller will respond to the I/O space accesses. |

### 11.1.4 PCISTS—PCI Status Register (LAN Controller—B1:D8:F0)

Offset Address: 06h–07h
Default Value: 0290h
Attribute: RO, R/WC
Size: 16 bits

***Note:*** For the writable bits, software must write a 1 to clear bits that are set. Writing a 0 to the bit has no effect.

| Bit | Description |
|---|---|
| 15 | **Detected Parity Error (DPE)** — R/WC.<br>0 = Parity error Not detected.<br>1 = The Chipset's integrated LAN controller has detected a parity error on the PCI bus (will be set even if Parity Error Response is disabled in the PCI Command register). |
| 14 | **Signaled System Error (SSE)** — R/WC.<br>0 = Integrated LAN controller has **not** asserted SERR#<br>1 = The chipset's integrated LAN controller has asserted SERR#. SERR# can be routed to cause NMI, SMI#, or interrupt. |
| 13 | **Master Abort Status (RMA)** — R/WC.<br>0 = Master Abort **not** generated<br>1 = The chipset's integrated LAN controller (as a PCI master) has generated a master abort. |
| 12 | **Received Target Abort (RTA)** — R/WC.<br>0 = Target abort **not** received.<br>1 = The chipset's integrated LAN controller (as a PCI master) has received a target abort. |
| 11 | Signaled Target Abort (STA) — RO. Hardwired to 0. The device will not signal Target Abort. |
| 10:9 | DEVSEL# Timing Status (DEV_STS) — RO.<br>01h = Medium timing. |
| 8 | **Data Parity Error Detected (DPED)** — R/WC.<br>0 = Parity error **not** detected (conditions below are not met).<br>1 = All of the following three conditions have been met:<br>1. The LAN controller is acting as bus master<br>2. The LAN controller has asserted PERR# (for reads) or detected PERR# asserted (for writes)<br>3. The Parity Error Response bit in the LAN controller's PCI Command Register is set. |
| 7 | **Fast Back to Back Capable (FB2BC)** — RO. Hardwired to 1. The device can accept fast back-to-back transactions. |
| 6 | **User Definable Features (UDF)** — RO. Hardwired to 0. Not implemented. |
| 5 | **66 MHz Capable (66MHZ_CAP)** — RO. Hardwired to 0. The device does not support 66 MHz PCI. |
| 4 | **Capabilities List (CAP_LIST)** — RO.<br>0 = The EEPROM indicates that the integrated LAN controller does not support PCI Power Management.<br>1 = The EEPROM indicates that the integrated LAN controller supports PCI Power Management. |
| 3 | **Interrupt Status (INTS)** — RO. This bit indicates that an interrupt is pending. It is independent from the state of the Interrupt Enable bit in the command register. |



| Bit | Description |
|---|---|
| 2:0 | **Reserved** |

### 11.1.5 RID—Revision Identification Register (LAN Controller—B1:D8:F0)

Offset Address: 08h Attribute: RO
Default Value: See bit description Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Revision ID (RID)** — RO. This field is an 8-bit value that indicates the revision number for the integrated LAN controller. The three least significant bits in this register may be overridden by the ID and REV ID fields in the EEPROM. Refer to the *Intel® I/O Controller Hub 7 Family Specification Update* for the value of the Revision ID Register. |

### 11.1.6 SCC—Sub Class Code Register (LAN Controller—B1:D8:F0)

Offset Address: 0Ah Attribute: RO
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Sub Class Code (SCC) — RO**. This 8-bit value specifies the sub-class of the device as an Ethernet controller. |

### 11.1.7 BCC—Base-Class Code Register (LAN Controller—B1:D8:F0)

Offset Address: 0Bh Attribute: RO
Default Value: 02h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Base Class Code (BCC) — RO**. This 8-bit value specifies the base class of the device as a network controller. |

### 11.1.8 CLS—Cache Line Size Register (LAN Controller—B1:D8:F0)

Offset Address: 0Ch
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bit | Description |
|---|---|
| 7:5 | **Reserved** |
| 4:3 | **Cache Line Size (CLS)** — R/W.<br>00 = Memory Write and Invalidate (MWI) command will not be used by the integrated LAN controller.<br>01 = MWI command will be used with Cache Line Size set to 8 DWords (only set if a value of 08h is written to this register).<br>10 = MWI command will be used with Cache Line Size set to 16 DWords (only set if a value of 10h is written to this register).<br>11 = Invalid. MWI command will not be used. |
| 2:0 | **Reserved** |

### 11.1.9 PMLT—Primary Master Latency Timer Register (LAN Controller—B1:D8:F0)

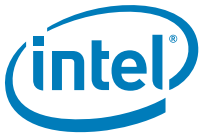
Offset Address: 0Dh
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bit | Description |
|---|---|
| 7:3 | **Master Latency Timer Count (MLTC)** — R/W. This field defines the number of PCI clock cycles that the integrated LAN controller may own the bus while acting as bus master. |
| 2:0 | **Reserved** |

### 11.1.10 HEADTYP—Header Type Register (LAN Controller—B1:D8:F0)

Offset Address: 0Eh
Default Value: 00h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7 | **Multi-Function Device (MFD) — RO**. Hardwired to 0 to indicate a single function device. |
| 6:0 | **Header Type (HTYPE) — RO**. This 7-bit field identifies the header layout of the configuration space as an Ethernet controller. |



### 11.1.11 CSR_MEM_BASE — CSR Memory-Mapped Base Address Register (LAN Controller—B1:D8:F0)

Offset Address: 10h–13h
Attribute: R/W, RO
Default Value: 00000008h
Size: 32 bits

***Note:*** The chipset's integrated LAN controller requires one BAR for memory mapping. Software determines which BAR (memory or I/O) is used to access the LAN controller's CSR registers.

| Bit | Description |
|---|---|
| 31:12 | **Base Address (MEM_ADDR)** — R/W. This field contains the upper 20 bits of the base address provides 4 KB of memory-Mapped space for the LAN controller's Control/Status registers. |
| 11:4 | **Reserved** |
| 3 | **Prefetchable (MEM_PF) — RO**. Hardwired to 0 to indicate that this is not a pre-fetchable memory-Mapped address range. |
| 2:1 | **Type (MEM_TYPE) — RO**. Hardwired to 00b to indicate the memory-Mapped address range may be located anywhere in 32-bit address space. |
| 0 | **Memory-Space Indicator (MEM_SPACE) — RO**. Hardwired to 0 to indicate that this base address maps to memory space. |

### 11.1.12 CSR_IO_BASE — CSR I/O-Mapped Base Address Register (LAN Controller—B1:D8:F0)

Offset Address: 14h–17h
Attribute: R/W, RO
Default Value: 00000001h
Size: 32 bits

***Note:*** The chipset's integrated LAN controller requires one BAR for memory mapping. Software determines which BAR (memory or I/O) is used to access the LAN controller's CSR registers.

| Bit | Description |
|---|---|
| 31:16 | **Reserved** |
| 15:6 | **Base Address (IO_ADDR)**— R/W. This field provides 64 bytes of I/O-Mapped address space for the LAN controller's Control/Status registers. |
| 5:1 | **Reserved** |
| 0 | **I/O Space Indicator (IO_SPACE) — RO**. Hardwired to 1 to indicate that this base address maps to<br>I/O space. |

### 11.1.13 SVID — Subsystem Vendor Identification (LAN Controller—B1:D8:F0)

Offset Address: 2Ch–2D Attribute: RO
Default Value: 0000h Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Subsystem Vendor ID (SVID) — RO**. See Section 11.1.14 for details. |

### 11.1.14 SID — Subsystem Identification (LAN Controller—B1:D8:F0)

Offset Address: 2Eh–2Fh Attribute: RO
Default Value: 0000h Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Subsystem ID (SID) — RO**. |

***Note:*** The chipset's integrated LAN controller provides support for configurable Subsystem ID and Subsystem Vendor ID fields. After reset, the LAN controller automatically reads addresses Ah through Ch, and 23h of the EEPROM. The LAN controller checks bits 15:13 in the EEPROM word Ah, and functions according to Table 11-112.

**Table 11-112.Configuration of Subsystem ID and Subsystem Vendor ID via EEPROM**

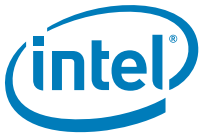
| Bits 15:14 | Bit 13 | Device ID[1] | Vendor ID | Revision ID[2] | Subsystem ID | Subsystem Vendor ID |
|---|---|---|---|---|---|---|
| 11b, 10b, 00b | X | 1051h | 8086h | 00h | 0000h | 0000h |
| 01b | 0b | Word 23h | 8086h | 00h | Word Bh | Word Ch |
| 01b | 1b | Word 23h | Word Ch | 80h + Word Ah, bits 10:8 | Word Bh | Word Ch |

**NOTES:**
1. The Device ID is loaded from Word 23h only if the value of Word 23h is not 0000h or FFFFh
2. The Revision ID is subject to change according to the silicon stepping.

### 11.1.15CAP_PTR — Capabilities Pointer (LAN Controller—B1:D8:F0)

Offset Address:34hAttribute: RO
Default Value: DCh Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Capabilities Pointer (CAP_PTR) — RO**. Hardwired to DCh to indicate the offset within configuration space for the location of the Power Management registers. |



### 11.1.16 INT_LN — Interrupt Line Register (LAN Controller—B1:D8:F0)

Offset Address: 3Ch Attribute: R/W
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Interrupt Line (INT_LN)** — R/W. This field identifies the system interrupt line to which the LAN controller's PCI interrupt request pin (as defined in the Interrupt Pin Register) is routed. |

### 11.1.17 INT_PN — Interrupt Pin Register (LAN Controller—B1:D8:F0)

Offset Address: 3Dh Attribute: RO
Default Value: 01h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Interrupt Pin (INT_PN) — RO**. Hardwired to 01h to indicate that the LAN controller's interrupt request is connected to PIRQA#. However, in the Chipset implementation, when the LAN controller interrupt is generated PIRQE# will go active, not PIRQA#. Note that if the PIRQE# signal is used as a GPI, the external visibility will be lost (though PIRQE# will still go active internally). |

### 11.1.18 MIN_GNT — Minimum Grant Register (LAN Controller—B1:D8:F0)

Offset Address: 3Eh Attribute: RO
Default Value: 08h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Minimum Grant (MIN_GNT) — RO**. This field indicates the amount of time (in increments of 0.25 µs) that the LAN controller needs to retain ownership of the PCI bus when it initiates a transaction. |

### 11.1.19 MAX_LAT — Maximum Latency Register (LAN Controller—B1:D8:F0)

Offset Address: 3Fh Attribute: RO
Default Value: 38h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Maximum Latency (MAX_LAT) — RO**. This field defines how often (in increments of 0.25 µs) the LAN controller needs to access the PCI bus. |

### 11.1.20 CAP_ID — Capability Identification Register (LAN Controller—B1:D8:F0)

Offset Address: DCh
Default Value: 01h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Capability ID (CAP_ID) — RO**. Hardwired to 01h to indicate that the chipset's integrated LAN controller supports PCI power management. |

### 11.1.21 NXT_PTR — Next Item Pointer (LAN Controller—B1:D8:F0)

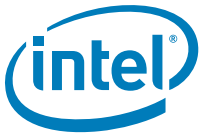
Offset Address: DDh
Default Value: 00h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Next Item Pointer (NXT_PTR) — RO**. Hardwired to 00b to indicate that power management is the last item in the capabilities list. |

### 11.1.22 PM_CAP — Power Management Capabilities (LAN Controller—B1:D8:F0)

Offset Address: DEh–DFh
Default Value: 7E21h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:11 | **PME Support (PME_SUP) — RO**. Hardwired to 11111b. This 5-bit field indicates the power states in which the LAN controller may assert PME#. The LAN controller supports wake-up in all power states. |
| 10 | **D2 Support (D2_SUP) — RO**. Hardwired to 1 to indicate that the LAN controller supports the D2 power state. |
| 9 | **D1 Support (D1_SUP) — RO**. Hardwired to 1 to indicate that the LAN controller supports the D1 power state. |
| 8:6 | **Auxiliary Current (AUX_CUR) — RO**. Hardwired to 000b to indicate that the LAN controller implements the Data registers. The auxiliary power consumption is the same as the current consumption reported in the D3 state in the Data register. |
| 5 | **Device Specific Initialization (DSI) — RO**. Hardwired to 1 to indicate that special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it. DSI is required for the LAN controller after D3-to-D0 reset. |
| 4 | **Reserved** |
| 3 | **PME Clock (PME_CLK) — RO**. Hardwired to 0 to indicate that the LAN controller does not require a clock to generate a power management event. |
| 2:0 | **Version (VER) — RO**. Hardwired to 010b to indicate that the LAN controller complies with of the *PCI Power Management Specification, Revision 1.1*. |



### 11.1.23 PMCSR — Power Management Control/ Status Register (LAN Controller—B1:D8:F0)

Offset Address: E0h–E1h Attribute: RO, R/W, R/WC
Default Value: 0000h Size: 16 bits

| Bit | Description |
|---|---|
| 15 | **PME Status (PME_STAT)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it. This also deasserts the PME# signal and clears the PME status bit in the Power Management Driver Register. When the PME# signal is enabled, the PME# signal reflects the state of the PME status bit.<br>1 = Set upon occurrence of a wake-up event, independent of the state of the PME enable bit. |
| 14:13 | **Data Scale (DSCALE)** — RO. This field indicates the data register scaling factor. It equals 10b for registers 0 through 8 and 00b for registers nine through fifteen, as selected by the “Data Select” field. |
| 12:9 | **Data Select (DSEL)** — R/W. This field is used to select which data is reported through the Data register and Data Scale field. |
| 8 | **PME Enable (PME_EN)** — R/W. This bit enables the chipset’s integrated LAN controller to assert PME#.<br>0 = The device will not assert PME#.<br>1 = Enable PME# assertion when PME Status is set. |
| 7:5 | **Reserved** |
| 4 | **Dynamic Data (DYN_DAT) — RO**. Hardwired to 0 to indicate that the device does not support the ability to monitor the power consumption dynamically. |
| 3:2 | **Reserved** |
| 1:0 | **Power State (PWR_ST)** — R/W. This 2-bit field is used to determine the current power state of the integrated LAN controller, and to put it into a new power state. The definition of the field values is as follows:<br>00 = D0<br>01 = D1<br>10 = D2<br>11 = D3 |

### 11.1.24 PCIDATA — PCI Power Management Data Register (LAN Controller—B1:D8:F0)

Offset Address: E3h
Default Value: 00h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Power Management Data (PWR_MGT) — RO**. State dependent power consumption and heat dissipation data. |

The data register is an 8-bit read only register that provides a mechanism for the chipset's integrated LAN controller to report state dependent maximum power consumption and heat dissipation. The value reported in this register depends on the value written to the Data Select field in the PMCSR register. The power measurements defined in this register have a dynamic range of 0 W to 2.55 W with 0.01 W resolution, scaled according to the Data Scale field in the PMCSR. The structure of the Data Register is given in Table 11-113.

**Table 11-113.Data Register Structure**

| Data Select | Data Scale | Data Reported |
|---|---|---|
| 0 | 2 | D0 Power Consumption |
| 1 | 2 | D1 Power Consumption |
| 2 | 2 | D2 Power Consumption |
| 3 | 2 | D3 Power Consumption |
| 4 | 2 | D0 Power Dissipated |
| 5 | 2 | D1 Power Dissipated |
| 6 | 2 | D2 Power Dissipated |
| 7 | 2 | D3 Power Dissipated |
| 8 | 2 | Common Function Power Dissipated |
| 9–15 | 0 | **Reserved** |

## 11.2 LAN Control / Status Registers (CSR) (LAN Controller—B1:D8:F0)

**Table 11-114.Chipset Integrated LAN Controller CSR Space Register Address Map**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00h–01h | SCB_STA | System Control Block Status Word | 0000h | R/WC, RO |
| 02h–03h | SCB_CMD | System Control Block Command Word | 0000h | R/W, WO |
| 04h–07h | SCB_GENPNT | System Control Block General Pointer | 0000 0000h | R/W |

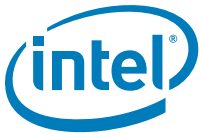


**Table 11-114.Chipset Integrated LAN Controller CSR Space Register Address Map**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 08h–0Bh | Port | PORT Interface | 0000 0000h | R/W (special) |
| 0Ch–0Dh | — | **Reserved** | — | — |
| 0Eh | EEPROM_CNTL | EEPROM Control | 00 | R/W, RO, WO |
| 0Fh | — | **Reserved** | — | — |
| 10h–13h | MDI_CNTL | Management Data Interface Control | 0000 0000h | R/W (special) |
| 14h–17h | REC_DMA_BC | Receive DMA Byte Count | 0000 0000h | RO |
| 18h | EREC_INTR | Early Receive Interrupt | 00h | R/W |
| 19–1Ah | FLOW_CNTL | Flow Control | 0000h | RO, R/W (special) |
| 1Bh | PMDR | Power Management Driver | 00h | R/WC |
| 1Ch | GENCNTL | General Control | 00h | R/W |
| 1Dh | GENSTA | General Status | 00h | RO |
| 1Eh | — | **Reserved** | — | — |
| 1Fh | SMB_PCI | SMB via PCI | 27h | R/W |
| 20h–3Ch | — | **Reserved** | — | — |

## 11.2.1 SCB_STA—System Control Block Status Word Register (LAN Controller—B1:D8:F0)

Offset Address: 00h–01h
Default Value: 0000h
Attribute: R/WC, RO
Size: 16 bits

The chipset's integrated LAN controller places the status of its Command Unit (CU) and Receive Unit (RC) and interrupt indications in this register for the processor to read.

| Bit | Description |
|---|---|
| 15 | **Command Unit (CU) Executed (CX)** — R/WC.<br>0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.<br>1 = Interrupt signaled because the CU has completed executing a command with its interrupt bit set. |
| 14 | **Frame Received (FR)** — R/WC.<br>0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.<br>1 = Interrupt signaled because the Receive Unit (RU) has finished receiving a frame. |

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>13</td><td>CU Not Active (CNA) — R/WC.<br>0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.<br>1 = The Command Unit left the Active state or entered the Idle state. There are 2 distinct states of the CU. When configured to generate CNA interrupt, the interrupt will be activated when the CU leaves the Active state and enters either the Idle or the Suspended state. When configured to generate CI interrupt, an interrupt will be generated only when the CU enters the Idle state.</td></tr>
<tr><td>12</td><td>Receive Not Ready (RNR) — R/WC.<br>0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.<br>1 = Interrupt signaled because the Receive Unit left the Ready state. This may be caused by an RU Abort command, a no resources situation, or set suspend bit due to a filled Receive Frame Descriptor.</td></tr>
<tr><td>11</td><td>Management Data Interrupt (MDI) — R/WC.<br>0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.<br>1 = Set when a Management Data Interface read or write cycle has completed. The management data interrupt is enabled through the interrupt enable bit (bit 29 in the Management Data Interface Control register in the CSR).</td></tr>
<tr><td>10</td><td>Software Interrupt (SWI) — R/WC.<br>0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.<br>1 = Set when software generates an interrupt.</td></tr>
<tr><td>9</td><td>Early Receive (ER) — R/WC.<br>0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.<br>1 = Indicates the occurrence of an Early Receive Interrupt.</td></tr>
<tr><td>8</td><td>Flow Control Pause (FCP) — R/WC.<br>0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.<br>1 = Indicates Flow Control Pause interrupt.</td></tr>
<tr><td>7:6</td><td>Command Unit Status (CUS) — RO.<br>00 = Idle<br>01 = Suspended<br>10 = LPQ (Low Priority Queue) active<br>11 = HPQ (High Priority Queue) active</td></tr>
<tr><td>5:2</td><td>Receive Unit Status (RUS) — RO.
<table>
<tr><th>Value</th><th>Status</th><th>Value</th><th>Status</th></tr>
<tr><td>0000b</td><td>Idle</td><td>1000b</td><td>Reserved</td></tr>
<tr><td>0001b</td><td>Suspended</td><td>1001b</td><td>Suspended with no more RBDs</td></tr>
<tr><td>0010b</td><td>No Resources</td><td>1010b</td><td>No resources due to no more RBDs</td></tr>
<tr><td>0011b</td><td>Reserved</td><td>1011b</td><td>Reserved</td></tr>
<tr><td>0100b</td><td>Ready</td><td>1100b</td><td>Ready with no RBDs present</td></tr>
<tr><td>0101b</td><td>Reserved</td><td>1101b</td><td>Reserved</td></tr>
<tr><td>0110b</td><td>Reserved</td><td>1110b</td><td>Reserved</td></tr>
<tr><td>0111b</td><td>Reserved</td><td>1111b</td><td>Reserved</td></tr>
</table></td></tr>
</table>

| Bit | Description |
|---|---|
| 1:0 | **Reserved** |

## 11.2.2 SCB_CMD—System Control Block Command Word Register (LAN Controller—B1:D8:F0)

Offset Address: 02h–03h
Default Value: 0000h
Attribute: R/W, WO
Size: 16 bits

The processor places commands for the Command and Receive units in this register. Interrupts are also acknowledged in this register.

| Bit | Description |
|---|---|
| 15 | **CX Mask (CX_MSK)** — R/W.<br>0 = Interrupt not masked.<br>1 = Disable the generation of a CX interrupt. |
| 14 | **FR Mask (FR_MSK)** — R/W.<br>0 = Interrupt not masked.<br>1 = Disable the generation of an FR interrupt. |
| 13 | **CNA Mask (CNA_MSK)** — R/W.<br>0 = Interrupt not masked.<br>1 = Disable the generation of a CNA interrupt. |
| 12 | **RNR Mask (RNR_MSK)** — R/W.<br>0 = Interrupt not masked.<br>1 = Disable the generation of an RNR interrupt. |
| 11 | **ER Mask (ER_MSK)** — R/W.<br>0 = Interrupt not masked.<br>1 = Disable the generation of an ER interrupt. |
| 10 | **FCP Mask (FCP_MSK)** — R/W.<br>0 = Interrupt not masked.<br>1 = Disable the generation of an FCP interrupt. |
| 9 | **Software Generated Interrupt (SI)** — WO.<br>0 = No Effect.<br>1 = Setting this bit causes the LAN controller to generate an interrupt. |
| 8 | **Interrupt Mask (IM)** — R/W. This bit enables or disables the LAN controller's assertion of the INTA# signal. This bit has higher precedence that the Specific Interrupt Mask bits and the SI bit.<br>0 = Enable the assertion of INTA#.<br>1 = Disable the assertion of INTA#. |

| Bit | Description |
|---|---|
| 7:4 | **Command Unit Command (CUC)** — R/W. Valid values are listed below. All other values are **Reserved**.<br>0000 = **NOP:** Does not affect the current state of the unit.<br>0001 = **CU Start**: Start execution of the first command on the CBL. A pointer to the first CB of the CBL should be placed in the SCB General Pointer before issuing this command. The CU Start command should only be issued when the CU is in the Idle or Suspended states (not when the CU is in the active state), and all of the previously issued Command Blocks have been processed and completed by the CU. Sometimes it is only possible to determine that all Command Blocks are completed by checking that the Complete bit is set in all previously issued Command Blocks.<br>0010 = **CU Resume:** Resume operation of the Command unit by executing the next command. This command will be ignored if the CU is idle.<br>0011 = **CU HPQ Start:** Start execution of the first command on the high priority CBL. A pointer to the first CB of the HPQ CBL should be placed in the SCB General POinter before issuing this command.<br>0100 = **Load Dump Counters Address:** Indicates to the device where to write dump data when using the Dump Statistical Counters or Dump and Reset Statistical Counters commands. This command must be executed at least once before any usage of the Dump Statistical Counters or Dump and Reset Statistical Counters commands. The address of the dump area must be placed in the General Pointer register.<br>0101 = **Dump Statistical Counters:** Tells the device to dump its statistical counters to the area designated by the Load Dump Counters Address command.<br>0110 = **Load CU Base:** The device's internal CU Base Register is loaded with the value in the CSB General Pointer.<br>0111 = **Dump and Reset Statistical Counters:** Indicates to the device to dump its statistical counters to the area designated by the Load Dump Counters Address command, and then to clear these counters.<br>1010 = **CU Static Resume:** Resume operation of the Command unit by executing the next command. This command will be ignored if the CU is idle. This command should be used only when the CU is in the Suspended state and has no pending CU Resume commands.<br>1011 = **CU HPQ Resume:** Resume execution of the first command on the HPQ CBL. this command will be ignored if the HPQ was not started. |
| 3 | **Reserved** |

| Bit | Description |
|---|---|
| 2:0 | **Receive Unit Command (RUC)** — R/W. Valid values are:<br>000 = **NOP:** Does not affect the current state of the unit.<br>001 = **RU Start:** Enables the receive unit. The pointer to the RFA must be placed in the SCB General POinter before using this command. The device pre-fetches the first RFD and the first RBD (if in flexible mode) in preparation to receive incoming frames that pass its address filtering.<br>010 = **RU Resume:** Resume frame reception (only when in suspended state).<br>011 = **RCV DMA Redirect:** Resume the RCV DMA when configured to “Direct DMA Mode.” The buffers are indicated by an RBD chain which is pointed to by an offset stored in the General Pointer Register (this offset will be added to the RU Base).<br>100 = **RU Abort:** Abort RU receive operation immediately.<br>101 = **Load Header Data Size (HDS):** This value defines the size of the Header portion of the RFDs or Receive buffers. The HDS value is defined by the lower 14 bits of the SCB General Pointer, so bits 31:15 should always be set to 0’s when using this command. Once a Load HDS command is issued, the device expects only to find Header RFDs, or be used in “RCV Direct DMA mode” until it is reset. Note that the value of HDS should be an even, non-zero number.<br>110 = **Load RU Base:** The device’s internal RU Base Register is loaded with the value in the SCB General Pointer.<br>111 = **RBD Resume:** Resume frame reception into the RFA. This command should only be used when the RU is already in the “No Resources due to no RBDs” state or the “Suspended with no more RBDs” state. |

### 11.2.3 SCB_GENPNT—System Control Block General Pointer Register (LAN Controller—B1:D8:F0)

Offset Address: 04h–07h — Attribute: R/W
Default Value: 0000 0000h — Size: 32 bits

| Bit | Description |
|---|---|
| 15:0 | **SCB General Pointer** — R/W. The SCB General Pointer register is programmed by software to point to various data structures in main memory depending on the current SCB Command word. |

### 11.2.4 PORT—PORT Interface Register (LAN Controller—B1:D8:F0)

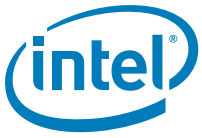
Offset Address: 08h–0Bh — Attribute: R/W (special)
Default Value: 0000 0000h — Size: 32 bits

The PORT interface allows the processor to reset the chipset’s internal LAN controller, or perform an internal self test. The PORT DWord may be written as a 32-bit entity, two 16-bit entities, or four
8-bit entities. The LAN controller will only accept the command after the high byte (offset 0Bh) is written; therefore, the high byte must be written last.

| Bit | Description |
|---|---|
| 31:4 | **Pointer Field (PORT_PTR)** — R/W (special). A 16-byte aligned address must be written to this field when issuing a Self-Test command to the PORT interface.The results of the Self Test will be written to the address specified by this field. |
| 3:0 | **PORT Function Selection (PORT_FUNC)** — R/W (special). Valid values are listed below. All other values are **reserved**.<br>0000 = **PORT Software Reset:** Completely resets the LAN controller (all CSR and PCI registers). This command should not be used when the device is active. If a PORT Software Reset is desired, software should do a Selective Reset (described below), wait for the PORT register to be cleared (completion of the Selective Reset), and then issue the PORT Software Reset command. Software should wait approximately 10 µs after issuing this command before attempting to access the LAN controller's registers again.<br>0001 = **Self Test:** The Self-Test begins by issuing an internal Selective Reset followed by a general internal self-test of the LAN controller. The results of the self-test are written to memory at the address specified in the Pointer field of this register. The format of the self-test result is shown in Table 11-115. After completing the self-test and writing the results to memory, the LAN controller will execute a full internal reset and will re-initialize to the default configuration. Self-Test does not generate an interrupt of similar indicator to the host processor upon completion.<br>0010 = **Selective Reset:** Sets the CU and RU to the Idle state, but otherwise maintains the current configuration parameters (RU and CU Base, HDSSize, Error Counters, Configure information and Individual/Multicast Addresses are preserved). Software should wait approximately 10 µs after issuing this command before attempting to access the LAN controller's registers again. |

### Table 11-115.Self-Test Results Format

| Bit | Description |
|---|---|
| 31:13 | **Reserved** |
| 12 | **General Self-Test Result (SELF_TST)** — R/W (special).<br>0 = Pass<br>1 = Fail |
| 11:6 | **Reserved** |
| 5 | **Diagnose Result (DIAG_RSLT)** — R/W (special). This bit provides the result of an internal diagnostic test of the Serial Subsystem.<br>0 = Pass<br>1 = Fail |
| 4 | **Reserved** |
| 3 | **Register Result (REG_RSLT)** — R/W (special). This bit provides the result of a test of the internal Parallel Subsystem registers.<br>0 = Pass<br>1 = Fail |
| 2 | **ROM Content Result (ROM_RSLT)** — R/W (special). This bit provides the result of a test of the internal microcode ROM.<br>0 = Pass<br>1 = Fail |
| 1:0 | **Reserved** |



### 11.2.5 EEPROM_CNTL—EEPROM Control Register (LAN Controller—B1:D8:F0)

Offset Address: 0Eh
Default Value: 00h
Attribute: RO, R/W, WO
Size: 8 bits

The EEPROM Control Register is a 16-bit field that enables a read from and a write to the external EEPROM.

| Bit | Description |
|---|---|
| 7:4 | **Reserved** |
| 3 | **EEPROM Serial Data Out (EEDO)** — RO. Note that this bit represents “Data Out” from the perspective of the EEPROM device. This bit contains the value read from the EEPROM when performing read operations. |
| 2 | **EEPROM Serial Data In (EEDI)** — WO. Note that this bit represents “Data In” from the perspective of the EEPROM device. The value of this bit is written to the EEPROM when performing write operations. |
| 1 | **EEPROM Chip Select (EECS)** — R/W.<br>0 = Drives the chipset’s EE_CS signal low to disable the EEPROM. this bit must be set to 0 for a minimum of 1 µs between consecutive instruction cycles.<br>1 = Drives the chipset’s EE_CS signal high, to enable the EEPROM. |
| 0 | **EEPROM Serial Clock (EESK)** — R/W. Toggling this bit clocks data into or out of the EEPROM. Software must ensure that this bit is toggled at a rate that meets the EEPROM component’s minimum clock frequency specification.<br>0 = Drives the chipset’s EE_SHCLK signal low.<br>1 = Drives the chipset’s EE_SHCLK signal high. |

### 11.2.6 MDI_CNTL—Management Data Interface (MDI) Control Register (LAN Controller—B1:D8:F0)

Offset Address: 10h–13h
Default Value: 0000 0000h
Attribute: R/W (special)
Size: 32 bits

The Management Data Interface (MDI) Control register is a 32-bit field and is used to read and write bits from the LAN Connect component. This register may be written as a 32-bit entity, two 16-bit entities, or four 8-bit entities. The LAN controller will only accept the command after the high byte (offset 13h) is written; therefore, the high byte must be written last.

| Bit | Description |
|---|---|
| 31:30 | These bits are **reserved** and should be set to 00b. |
| 29 | **Interrupt Enable** — R/W (special).<br>0 = Disable.<br>1 = Enables the LAN controller to assert an interrupt to indicate the end of an MDI cycle. |
| 28 | **Ready** — R/W (special).<br>0 = Expected to be reset by software at the same time the command is written.<br>1 = Set by the LAN controller at the end of an MDI transaction. |

| Bit | Description |
|---|---|
| 27:26 | **Opcode** — R/W (special). These bits define the opcode:<br>00 = **Reserved**<br>01 = MDI write<br>10 = MDI read<br>11 = **Reserved** |
| 25:21 | **LAN Connect Address** — R/W (special). This field of bits contains the LAN Connect address. |
| 20:16 | **LAN Connect Register Address** — R/W (special). This field contains the LAN Connect Register Address. |
| 15:0 | **Data** — R/W (special). In a write command, software places the data bits in this field, and the LAN controller transfers the data to the external LAN Connect component. During a read command, the LAN controller reads these bits serially from the LAN Connect, and software reads the data from this location. |

## 11.2.7 REC_DMA_BC—Receive DMA Byte Count Register (LAN Controller—B1:D8:F0)

Offset Address: 14h–17h Attribute: RO
Default Value: 0000 0000h Size: 32 bits

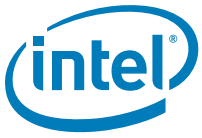
| Bit | Description |
|---|---|
| 31:0 | **Receive DMA Byte Count** — RO. This field keeps track of how many bytes of receive data have been passed into host memory via DMA. |

## 11.2.8 EREC_INTR—Early Receive Interrupt Register (LAN Controller—B1:D8:F0)

Offset Address: 18h Attribute: R/W
Default Value: 00h Size: 8 bits

The Early Receive Interrupt register allows the internal LAN controller to generate an early interrupt depending on the length of the frame. The LAN controller will generate an interrupt at the end of the frame regardless of whether or not Early Receive Interrupts are enabled.

***Note:*** It is recommended that software **not** use this register unless receive interrupt latency is a critical performance issue in that particular software environment. Using this feature may reduce receive interrupt latency, but will also result in the generation of more interrupts, which can degrade system efficiency and performance in some environments.



| Bit | Description |
|---|---|
| 7:0 | **Early Receive Count** — R/W. When some non-zero value ***x*** is programmed into this register, the LAN controller will set the ER bit in the SCB Status Word Register and assert INTA# when the byte count indicates that there are ***x*** qwords remaining to be received in the current frame (based on the Type/Length field of the received frame). No Early Receive interrupt will be generated if a value of 00h (the default value) is programmed into this register. |

## 11.2.9 FLOW_CNTL—Flow Control Register (LAN Controller—B1:D8:F0)

Offset Address: 19h–1Ah
Default Value: 0000h
Attribute: RO, R/W (special)
Size: 16 bits

| Bit | Description |
|---|---|
| 15:13 | **Reserved** |
| 12 | **FC Paused Low** — RO.<br>0 = Cleared when the FC timer reaches 0, or a Pause frame is received.<br>1 = Set when the LAN controller receives a Pause Low command with a value greater than 0. |
| 11 | **FC Paused** — RO.<br>0 = Cleared when the FC timer reaches 0.<br>1 = Set when the LAN controller receives a Pause command regardless of its cause (FIFO reaching Flow Control Threshold, fetching a Receive Frame Descriptor with its Flow Control Pause bit set, or software writing a 1 to the Xoff bit). |
| 10 | **FC Full** — RO.<br>0 = Cleared when the FC timer reaches 0.<br>1 = Set when the LAN controller sends a Pause command with a value greater than 0. |
| 9 | **Xoff** — R/W (special). This bit should only be used if the LAN controller is configured to operate with IEEE frame-based flow control.<br>0 = This bit can only be cleared by writing a 1 to the Xon bit (bit 8 in this register).<br>1 = Writing a 1 to this bit forces the Xoff request to 1 and causes the LAN controller to behave as if the FIFO extender is full. This bit will also be set to 1 when an Xoff request due to an “RFD Xoff” bit. |

| Bit | Description |
|---|---|
| 8 | **Xon** — WO. This bit should only be used if the LAN controller is configured to operate with IEEE frame-based flow control.<br>0 = This bit always returns 0 on reads.<br>1 = Writing a 1 to this bit resets the Xoff request to the LAN controller, clearing bit 9 in this register. |
| 7:3 | **Reserved** |
| 2:0 | **Flow Control Threshold** — R/W. The LAN controller can generate a Flow Control Pause frame when its Receive FIFO is almost full. The value programmed into this field determines the number of bytes still available in the Receive FIFO when the Pause frame is generated. |

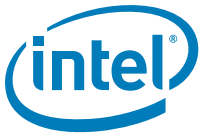
| Bits 2:0 | Free Bytes in RX FIFO | Comment |
|---|---|---|
| 000b | 0.50 KB | Fast system (recommended default) |
| 001b | 1.00 KB | |
| 010b | 1.25 KB | |
| 011b | 1.50 KB | |
| 100b | 1.75 KB | |
| 101b | 2.00 KB | |
| 110b | 2.25 KB | |
| 111b | 2.50 KB | Slow system |

## 11.2.10 PMDR—Power Management Driver Register (LAN Controller—B1:D8:F0)

Offset Address: 1Bh
Default Value: 00h
Attribute: R/WC
Size: 8 bits

The chipset's internal LAN controller provides an indication in the PMDR that a wake-up event has occurred.

| Bit | Description |
|---|---|
| 7 | **Link Status Change Indication** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = The link status change bit is set following a change in link status. |
| 6 | **Magic Packet** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit is set when a Magic Packet is received regardless of the Magic Packet wake-up disable bit in the configuration command and the PME Enable bit in the Power Management Control/ Status Register. |
| 5 | **Interesting Packet** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit is set when an "interesting" packet is received. Interesting packets are defined by the LAN controller packet filters. |



| Bit | Description |
|---|---|
| 4:3 | **Reserved** |
| 2 | **ASF Enabled** — RO. This bit is set to 1 when the LAN controller is in ASF mode. |
| 1 | **TCO Request** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit is set to 1b when the LAN controller is busy with TCO activity. |
| 0 | **PME Status** — R/WC. This bit is a reflection of the PME Status bit in the Power Management Control/Status Register (PMCSR).<br>0 = Software clears this bit by writing a 1 to it.This also clears the PME Status bit in the PMCSR and deasserts the PME signal.<br>1 = Set upon a wake-up event, independent of the PME Enable bit. |

## 11.2.11 GENCNTL—General Control Register (LAN Controller—B1:D8:F0)

Offset Address: 1Ch Attribute: R/W
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:4 | **Reserved**. These bits should be set to 0000b. |
| 3 | **LAN Connect Software Reset** — R/W.<br>0 = Cleared by software to begin normal LAN Connect operating mode. Software must not attempt to access the LAN Connect interface for at least 1ms after clearing this bit.<br>1 = Software can set this bit to force a reset condition on the LAN Connect interface. |
| 2 | **Reserved**. This bit should be set to 0. |
| 1 | **Deep Power-Down on Link Down Enable** — R/W.<br>0 = Disable<br>1 = Enable. The chipset's internal LAN controller may enter a deep power-down state (sub-3 mA) in the D2 and D3 power states while the link is down. In this state, the LAN controller does not keep link integrity. This state is not supported for point-to-point connection of two end stations. |
| 0 | **Reserved** |

## 11.2.12 GENSTA—General Status Register (LAN Controller—B1:D8:F0)

Offset Address: 1Dh Attribute: RO
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:3 | **Reserved** |

| | |
|---|---|
| 2 | **Duplex Mode** — RO. This bit indicates the wire duplex mode.<br>0 = Half duplex<br>1 = Full duplex |
| 1 | **Speed** — RO. This bit indicates the wire speed.<br>0 = 10 Mb/s<br>1 = 100 Mb/s |
| 0 | **Link Status Indication** — RO. This bit indicates the status of the link.<br>0 = Invalid<br>1 = Valid |

## 11.2.13 SMB_PCI—SMB via PCI Register (LAN Controller—B1:D8:F0)

Offset Address: 1Fh Attribute: R/W, RO
Default Value: 27h Size: 8 bits

Software asserts SREQ when it wants to isolate the PCI-accessible SMBus to the ASF registers/commands. It waits for SGNT to be asserted. At this point SCLI, SDAO, SCLO, and SDAI can be toggled/read to force ASF controller SMBus transactions without affecting the external SMBus. After all operations are completed, the bus is returned to idle (SCLO=1b,SDAO=1b, SCLI=1b, SDAI=1b), SREQ is released (written 0b). Then SGNT goes low to indicate released control of the bus. The logic in the ASF controller only asserts or deasserts SGNT at times when it determines that it is safe to switch (all SMBuses that are switched in/out are idle).

When in isolation mode (SGNT=1), software can access the Chipset SMBus slaves that allow configuration without affecting the external SMBus. This includes configuration register accesses and ASF command accesses. However, this capability is not available to the external TCO controller. When SGNT=0, the bit-banging and reads are reflected on the main SMBus and the PCISML_SDA0, PCISML_SCL0 read only bits.

| Bit | Description |
|---|---|
| 7:6 | **Reserved** |
| 5 | **PCISML_SCLO** — RO. SMBus Clock from the ASF controller. |
| 4 | **PCISML_SGNT** — RO. SMBus Isolation Grant from the ASF controller. |
| 3 | **PCISML_SREQ** — R/W. SMBus Isolation Request to the ASF controller. |
| 2 | **PCISML_SDAO** — RO. SMBus Data from the ASF controller. |
| 1 | **PCISML_SDAI** — R/W. SMBus Data to the ASF controller. |
| 0 | **PCISML_SCLI** — R/W. SMBus Clock to the ASF controller. |

## 11.2.14 Statistical Counters (LAN Controller—B1:D8:F0)

The chipset's integrated LAN controller provides information for network management statistics by providing on-chip statistical counters that count a variety of events associated with both transmit and receive. The counters are updated by the LAN controller when it completes the processing of a frame (that is, when it has completed transmitting a frame on the link or when it has completed receiving a frame). The

Statistical Counters are reported to the software on demand by issuing the Dump Statistical Counters command or Dump and Reset Statistical Counters command in the SCB Command Unit Command (CUC) field.

**Table 11-116.Statistical Counters (Sheet 1 of 2)**

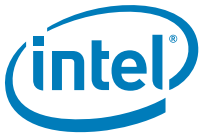
| ID | Counter | Description |
|---|---|---|
| 0 | Transmit Good Frames | This counter contains the number of frames that were transmitted properly on the link. It is updated only after the actual transmission on the link is completed, not when the frame was read from memory as is done for the Transmit Command Block status. |
| 4 | Transmit Maximum Collisions (MAXCOL) Errors | This counter contains the number of frames that were not transmitted because they encountered the configured maximum number of collisions. |
| 8 | Transmit Late Collisions (LATECOL) Errors | This counter contains the number of frames that were not transmitted since they encountered a collision later than the configured slot time. |
| 12 | Transmit Underrun Errors | A transmit underrun occurs because the system bus cannot keep up with the transmission. This counter contains the number of frames that were either not transmitted or retransmitted due to a transmit DMA underrun. If the LAN controller is configured to retransmit on underrun, this counter may be updated multiple times for a single frame. |
| 16 | Transmit Lost Carrier Sense (CRS) | This counter contains the number of frames that were transmitted by the LAN controller despite the fact that it detected the de-assertion of CRS during the transmission. |
| 20 | Transmit Deferred | This counter contains the number of frames that were deferred before transmission due to activity on the link. |
| 24 | Transmit Single Collisions | This counter contains the number of transmitted frames that encountered one collision. |
| 28 | Transmit Multiple Collisions | This counter contains the number of transmitted frames that encountered more than one collision. |
| 32 | Transmit Total Collisions | This counter contains the total number of collisions that were encountered while attempting to transmit. This count includes late collisions and frames that encountered MAXCOL. |
| 36 | Receive Good Frames | This counter contains the number of frames that were received properly from the link. It is updated only after the actual reception from the link is completed and all the data bytes are stored in memory. |
| 40 | Receive CRC Errors | This counter contains the number of aligned frames discarded because of a CRC error. This counter is updated, if needed, regardless of the Receive Unit state. The Receive CRC Errors counter is mutually exclusive of the Receive Alignment Errors and Receive Short Frame Errors counters. |
| 44 | Receive Alignment Errors | This counter contains the number of frames that are both misaligned (for example, CRS de-asserts on a non-octal boundary) and contain a CRC error. The counter is updated, if needed, regardless of the Receive Unit state. The Receive Alignment Errors counter is mutually exclusive of the Receive CRC Errors and Receive Short Frame Errors counters. |

**Table 11-116.Statistical Counters (Sheet 2 of 2)**

| ID | Counter | Description |
|---|---|---|
| 48 | Receive Resource Errors | This counter contains the number of good frames discarded due to unavailability of resources. Frames intended for a host whose Receive Unit is in the No Resources state fall into this category. If the LAN controller is configured to Save Bad Frames and the status of the received frame indicates that it is a bad frame, the Receive Resource Errors counter is not updated. |
| 52 | Receive Overrun Errors | This counter contains the number of frames known to be lost because the local system bus was not available. If the traffic problem persists for more than one frame, the frames that follow the first are also lost; however, because there is no lost frame indicator, they are not counted. |
| 56 | Receive Collision Detect (CDT) | This counter contains the number of frames that encountered collisions during frame reception. |
| 60 | Receive Short Frame Errors | This counter contains the number of received frames that are shorter than the minimum frame length. The Receive Short Frame Errors counter is mutually exclusive to the Receive Alignment Errors and Receive CRC Errors counters. A short frame will always increment only the Receive Short Frame Errors counter. |
| 64 | Flow Control Transmit Pause | This counter contains the number of Flow Control frames transmitted by the LAN controller. This count includes both the Xoff frames transmitted and Xon (PAUSE(0)) frames transmitted. |
| 68 | Flow Control Receive Pause | This counter contains the number of Flow Control frames received by the LAN controller. This count includes both the Xoff frames received and Xon (PAUSE(0)) frames received. |
| 72 | Flow Control Receive Unsupported | This counter contains the number of MAC Control frames received by the LAN controller that are not Flow Control Pause frames. These frames are valid MAC control frames that have the predefined MAC control Type value and a valid address but has an unsupported opcode. |
| 76 | Receive TCO Frames | This counter contains the number of TCO packets received by the LAN controller. |
| 78 | Transmit TCO Frames | This counter contains the number of TCO packets transmitted. |

The Statistical Counters are initially set to 0 by the chipset's integrated LAN controller after reset. They cannot be preset to anything other than 0. The LAN controller increments the counters by internally reading them, incrementing them and writing them back. This process is invisible to the processor and PCI bus. In addition, the counters adhere to the following rules:

- The counters are wrap-around counters. After reaching FFFFFFFFh the counters wrap around to 0.
- The LAN controller updates the required counters for each frame. It is possible for more than one counter to be updated as multiple errors can occur in a single frame.
- The counters are 32 bits wide and their behavior is fully compatible with the IEEE 802.1 standard. The LAN controller supports all mandatory and recommend statistics functions through the status of the receive header and directly through these Statistical Counters.



The processor can access the counters by issuing a Dump Statistical Counters SCB command. This provides a “snapshot”, in main memory, of the internal LAN controller statistical counters. The LAN controller supports 21 counters. The dump could consist of the either 16, 19, or all 21 counters, depending on the status of the Extended Statistics Counters and TCO Statistics configuration bits in the Configuration command.

## 11.3 ASF Configuration Registers (LAN Controller—B1:D8:F0)

The ASF registers in this table are accessible through the Chipset SMBus slave interface.

**Table 11-117.ASF Register Address Map**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| E0h | ASF_RID | ASF Revision Identification | ECh | RO |
| E1h | SMB_CNTL | SMBus Control | 40h | R/W |
| E2h | ASF_CNTL | ASF Control | 00h | R/W, RO |
| E3h | ASF_CNTL_EN | ASF Control Enable | 00h | R/W |
| E4h | ENABLE | Enable | 00h | R/W |
| E5h | APM | APM | 08h | R/W |
| E6h–E7h | — | **Reserved** | — | — |
| E8h | WTIM_CONF | Watchdog Timer Configuration | 00h | R/W |
| E9h | HEART_TIM | Heartbeat Timer | 02h | R/W |
| EAh | RETRAN_INT | Retransmission Interval | 02h | R/W |
| EBh | RETRAN_PCL | Retransmission Packet Count Limit | 03h | R/W |
| ECh | ASF_WTIM1 | ASF Watchdog Timer 1 | 01h | R/W |
| EDh | ASF_WTIM2 | ASF Watchdog Timer 2 | 00h | R/W |
| F0h | PET_SEQ1 | PET Sequence 1 | 00h | R/W |
| F1h | PET_SEQ2 | PET Sequence 2 | 00h | R/W |
| F2h | STA | Status | 40h | R/W |
| F3h | FOR_ACT | Forced Actions | 02h | R/W |
| F4h | RMCP_SNUM | RMCP Sequence Number | 00h | R/W |
| F5h | SP_MODE | Special Modes | x0h | R/WC, RO |
| F6h | INPOLL_TCONF | Inter-Poll Timer Configuration | 10h | R/W |
| F7h | PHIST_CLR | Poll History Clear | 00h | R/WC |
| F8h | PMSK1 | Polling Mask 1 | XXh | R/W |
| F9h | PMSK2 | Polling Mask 2 | XXh | R/W |
| FAh | PMSK3 | Polling Mask 3 | XXh | R/W |
| FBh | PMSK4 | Polling Mask 4 | XXh | R/W |
| FCh | PMSK5 | Polling Mask 5 | XXh | R/W |

**Table 11-117.ASF Register Address Map**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| FDh | PMSK6 | Polling Mask 6 | XXh | R/W |
| FEh | PMSK7 | Polling Mask 7 | XXh | R/W |
| FFh | PMSK8 | Polling Mask 8 | XXh | R/W |

## 11.3.1 ASF_RID—ASF Revision Identification Register (LAN Controller—B1:D8:F0)

Offset Address: E0h Attribute: RO
Default Value: ECh Size: 8 bits

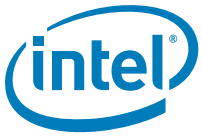
| Bit | Description |
|---|---|
| 7:3 | **ASF ID — RO**. Hardwired to 11101 to identify the ASF controller. |
| 2:0 | **ASF Silicon Revision — RO**. This field provides the silicon revision. |

## 11.3.2 SMB_CNTL—SMBus Control Register (LAN Controller—B1:D8:F0)

Offset Address: E1h Attribute: R/W
Default Value: 40h Size: 8 bits

This register is used to control configurations of the SMBus ports.

| Bit | Description |
|---|---|
| 7 | **SMBus Remote Control ASF Enable (SMB_RCASF) — R/W**.<br>0 = Legacy descriptors and operations are used.<br>1 = ASF descriptors and operations are used. |
| 6 | **SMBus ARP Enable (SMB_ARPEN) — R/W**.<br>0 = Disable.<br>1 = ASF enables the SMBus ARP protocol. |
| 5:4 | **Reserved** |
| 3 | **SMBus Drive Low (SMB_DRVLO) — R/W**.<br>0 = ASF will not drive the main SMBus signals low while PWR_GOOD = 0.<br>1 = ASF will drive the main SMBus signals low while PWR_GOOD = 0. |
| 2:0 | **Reserved** |



### 11.3.3 ASF_CNTL—ASF Control Register (LAN Controller—B1:D8:F0)

Offset Address: E2h
Default Value: 00h
Attribute: R/W, RO
Size: 8 bits

This register contains enables for special modes and SOS events. CTL_PWRLS should be set if ASF should be expecting a power loss due to software action. Otherwise, an EEPROM reload will happen when the power is lost.

| Bit | Description |
|---|---|
| 7 | **SMBus Hang SOS Enable (CTL_SMBHG) — R/W.**<br>0 = Disable<br>1 = Enables SMBus Hang SOS to be sent. |
| 6 | **Watchdog SOS Enable (CTL_WDG) — R/W.**<br>0 = Disable.<br>1 = Enables Watchdog SOS to be sent. |
| 5 | **Link Loss SOS Enable (CTL_LINK) — R/W.**<br>0 = Disable.<br>1 = Enables Link Loss SOS to be sent. |
| 4 | **OS Hung Status (CTL_OSHUNG) — RO.**<br>1 = This bit will be set to 1 when ASF has detected a Watchdog Expiration.<br>**NOTE:** This condition is only clearable by a PCI RST# assertion (system reset). |
| 3 | **Power-Up SOS Enable (CTL_PWRUP) — R/W.**<br>0 = Disable.<br>1 = Enables Power-Up SOS to be sent. |
| 2 | **Reserved** |
| 1 | **Receive ARP Enable (CTL_RXARP) — R/W.** The LAN controller interface provides a mode where all packets can be requested.<br>0 = Disable.<br>1 = Enable. ASF requests all packets when doing a Receive Enable. This is necessary in LAN controller to get ARP packets.<br>**NOTE:** Changes to this bit will not take effect until the next Receive Enable command to the LAN. |
| 0 | **Power Loss OK (CTL_PWRLS) — R/W.**<br>0 = Power Loss will reload EEPROM<br>1 = Power Loss will not reload EEPROM |

### 11.3.4 ASF_CNTL_EN—ASF Control Enable Register (ASF Controller—B1:D8:F0)

Offset Address: E3h
Default Value: 00h
Attribute: R/W
Size: 8 bits

This register is used to enable global processing as well as polling. GLOBAL ENABLE controls all of the SMBus processing and packet creation.

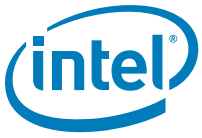
| Bit | Description |
|---|---|
| 7 | **Global Enable (CENA_ALL) — R/W.**<br>0 = Disable<br>1 = All control and polling enabled |
| 6 | **Receive Enable (CENA_RX) — R/W.**<br>0 = Disable<br>1 = TCO Receives enabled. |
| 5 | **Transmit Enable (CENA_TX) — R/W.**<br>0 = Disable<br>1 = SOS and RMCP Transmits enabled |
| 4 | **ASF Polling Enable (CENA_APOL) — R/W.**<br>0 = Disable<br>1 = Enable ASF Sensor Polling. |
| 3 | **Legacy Polling Enable (CENA_LPOL) — R/W.**<br>0 = Disable<br>1 = Enable Legacy Sensor Polling. |
| 2:0 | Number of Legacy Poll Devices (CENA_NLPOL) — R/W. This 3-bit value indicates how many of the eight possible polling descriptors are active.<br>000 = First polling descriptor is active.<br>001 = First two polling descriptors are active.<br>...<br>111 = Enables all eight descriptors. |

### 11.3.5 ENABLE—Enable Register (ASF Controller—B1:D8:F0)

Offset Address: E4h
Default Value: 00h
Attribute: R/W
Size: 8 bits

This register provides the mechanism to enable internal SOS operations and to enable the remote control functions.

| Bit | Description |
|---|---|
| 7 | **Enable OSHung ARPs (ENA_OSHARP) — R/W.**<br>0 = Disable<br>1 = ASF will request all packets when in a OSHung state. This allows ASF to receive ARP frames and respond as appropriate. |
| 6 | **State-based Security Destination Port Select (ENA_SB0298) — R/W.**<br>0 = State-based security will be honored on packets received on port 026Fh.<br>1 = Packets received on port 0298h will be honored. |



| Bit | Description |
|---|---|
| 5 | **PET VLAN Enable (ENA_VLAN) — R/W.**<br>0 = Disable<br>1 = Indicates a VLAN header for PET<br><br>**NOTE:** If this bit is set, the PET packet in EEPROM must have the VLAN tag within the packet. |
| 4 | **Reserved** |
| 3 | **System Power Cycle Enable (ENA_CYCLE) — R/W.**<br>0 = Disable<br>1 = Enables RMCP Power Cycle action. |
| 2 | **System Power-Down Enable (ENA_DWN) — R/W.**<br>0 = Disable<br>1 = Enables RMCP Power-Down action. |
| 1 | **System Power-Up Enable (ENA_UP) — R/W.**<br>0 = Disable<br>1 = Enables RMCP Power-Up action. |
| 0 | **System Reset Enable (ENA_RST) — R/W.**<br>0 = Disable<br>1 = Enables RMCP Reset action |

### 11.3.6 APM—APM Register (ASF Controller—B1:D8:F0)

Offset Address: E5h Attribute: R/W
Default Value: 08h Size: 8 bits

This register contains the configuration bit to disable state-based security.

| Bit | Description |
|---|---|
| 7:4 | **Reserved** |
| 3 | **Disable State-based Security (APM_DISSB) — R/W.**<br>0 = State-based security on OSHung is enabled.<br>1 = State-based security is disabled and actions are not gated by OSHung. |
| 2:0 | **Reserved** |

### 11.3.7 WTIM_CONF—Watchdog Timer Configuration Register (ASF Controller—B1:D8:F0)

Offset Address: E8h Attribute: R/W
Default Value: 00h Size: 8 bits

This register contains a single bit that enables the Watchdog timer. This bit is not intended to be accessed by software, but should be configured appropriately in the EEPROM location for this register default. The bit provides real-time control for enabling/disabling the Watchdog timer. When set the timer will count down. When cleared the counter will stop. Timer Start ASF SMBUS messages will set this bit. Timer Stop ASF SMBus transactions will clear this bit.

| Bit | Description |
|---|---|
| 7:1 | **Reserved** |
| 0 | **Timer Enable (WDG_ENA) — R/W.**<br>0 = Disable<br>1 = Enable Counter |

## 11.3.8 HEART_TIM—Heartbeat Timer Register (ASF Controller—B1:D8:F0)

Offset Address: E9h Attribute: R/W
Default Value: 02h Size: 8 bits

The HeartBeat Timer register implements the heartbeat timer. This defines the period of the heartbeats packets. It contains a down counting value when enabled and the time-out value when the counter is disabled. The timer can be configured and enabled in a single write.

***Note:*** The heartbeat timer controls the heartbeat status packet frequency. The timer is free-running and the configured time is only valid from one heartbeat to the next. When enabled by software, the next heartbeat may occur in any amount of time less than the configured time.

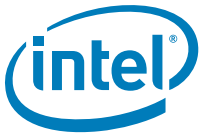
| Bit | Description |
|---|---|
| 7:1 | **Heartbeat Timer Value (HBT_VAL) — R/W**. Heartbeat timer load value in 10.7-second resolution. This field can only be written while the timer is disabled. (10.7 sec – 23 min range). Read as load value when HBT_ENA=0. Read as decrementing value when HBT_ENA=1. Timer resolution is 10.7 seconds. A value of 00h is invalid. |
| 0 | **Timer Enable (HBT_ENA) — R/W**.<br>0 = Disable<br>1 = Enable / Reset Counter |

## 11.3.9 RETRAN_INT—Retransmission Interval Register (ASF Controller—B1:D8:F0)

Offset Address: EAh Attribute: R/W
Default Value: 02h Size: 8 bits

This register implements the retransmission timer. This is the time between packet transmissions for multiple packets due to a SOS.

| Bit | Description |
|---|---|
| 7:1 | **Retransmit Timer Va**lue (RTM_VAL) — R/W. Retransmit timer load value 2.7 second resolution. This field is always writable (2.7 sec – 5.7 min range). Timer is accurate to +0 seconds, –0.336 seconds. Reads always show the **load value** (decrement value not shown). A value of 00h is invalid. |
| 0 | **Reserved** |



## 11.3.10 RETRAN_PCL—Retransmission Packet Count Limit

Register (ASF Controller—B1:D8:F0)

| | | | |
|---|---|---|---|
| Offset Address: | EBh | Attribute: | R/W |
| Default Value: | 03h | Size: | 8 bits |

This register defines the number of packets that are to be sent due to an SOS.

| Bit | Description |
|---|---|
| 7:0 | **Retransmission Packet Count Lim**it (RPC_VAL) — R/W. This field provides the number of packets to be sent for all SOS packets that require retransmissions. |

## 11.3.11 ASF_WTIM1—ASF Watchdog Timer 1 Register (ASF Controller—B1:D8:F0)

| | | | |
|---|---|---|---|
| Offset Address: | ECh | Attribute: | R/W |
| Default Value: | 01h | Size: | 8 bits |

This register is used to load the low byte of the timer. When read, it reports the decrementing value. This register is not intended to be written by software, but should be configured appropriately in the EEPROM location for this register default. Timer Start ASF SMBus transactions will load values into this register. Once the timer has expired (0000h), the timer will be disabled (EDG_ENA=0b) and the value in this register will remain at 00h until otherwise changed.

| Bit | Description |
|---|---|
| 7:0 | **ASF Watchdog T**imer 1 (AWD1_VAL) — R/W. This field provides the low byte of the ASF 1-second resolution timer. The timer is accurate to +0 seconds, –0.336 seconds. |

## 11.3.12 ASF_WTIM2—ASF Watchdog Timer 2 Register (ASF Controller—B1:D8:F0)

| | | | |
|---|---|---|---|
| Offset Address: | EDh | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bits |

This register is used to load the high byte of the timer. When read, it reports the decrementing value. This register is not intended to be written by software, but should be configured appropriately in the EEPROM location for this register default. Timer Start ASF SMBus transactions will load values into this register. Once the timer has expired (0000h), the timer will be disabled (EDG_ENA=0b) and the value in this register will remain at 00h until otherwise changed.

| Bit | Description |
|---|---|
| 7:0 | **ASF Watchdog Timer 2** (AWD2_VAL) — R/W. This field provides the high byte of the ASF 1-second resolution timer. The timer is accurate to +0 seconds, –0.336 seconds. |

### 11.3.13 PET_SEQ1—PET Sequence 1 Register (ASF Controller—B1:D8:F0)

Offset Address: F0h
Default Value: 00h
Attribute: R/W
Size: 8 bits

This register (low byte) holds the current value of the PET sequence number. This field is read/write-able through this register, and is also automatically incremented by the hardware when new PET packets are generated. By policy, software should not write to this register unless transmission is disabled.

| Bit | Description |
|---|---|
| 7:0 | **PET Sequence Byte** 1 (PSEQ1_VAL) — R/W. This field provides the low byte. |

### 11.3.14 PET_SEQ2—PET Sequence 2 Register (ASF Controller—B1:D8:F0)

Offset Address: F1h
Default Value: 00h
Attribute: R/W
Size: 8 bits

This register (high byte) holds the current value of the PET sequence number. This field is read/write-able through this register, and is also automatically incremented by the hardware when new PET packets are generated. By policy, software should not write to this register unless transmission is disabled.

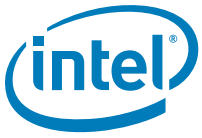
| Bit | Description |
|---|---|
| 7:0 | **PET Sequence Byte 2** (PSEQ2_VAL) — R/W. This field provides the high byte. |

### 11.3.15 STA—Status Register (ASF Controller—B1:D8:F0)

Offset Address: F2h
Default Value: 40h
Attribute: R/W
Size: 8 bits

This register gives status indication about several aspects of ASF.

| Bit | Description |
|---|---|
| 7 | **EEPROM Loading (STA_LOAD) — R/W**. EEPROM defaults are in the process of being loaded when this bit is a 1. |
| 6 | **EEPROM Invalid Checksum Indication (STA_ICRC) — R/W**. This bit should be read only after the EEC_LOAD bit is a 0.<br>0 = Valid<br>1 = Invalid checksum detected for ASF portion of the EEPROM. |
| 5:4 | **Reserved** |
| 3 | **Power Cycle Status (STA_CYCLE) — R/W**.<br>0 = Software clears this bit by writing a 1.<br>1 = This bit is set when a Power Cycle operation has been issued. |



| | |
|---|---|
| 2 | Power Down Status (STA_DOWN) — R/W.<br>0 = Software clears this bit by writing a 1<br>1 = This bit is set when a Power Down operation has been issued. |
| 1 | Power Up Status (STA_UP) — R/W.<br>0 = Software clears this bit by writing a 1<br>1 = This bit is set when a Power Up operation has been issued. |
| 0 | System Reset Status (STA_RST) — R/W.<br>0 = Software clears this bit by writing a 1<br>1 = This bit is set when a System Reset operation has been issued. |

## 11.3.16 FOR_ACT—Forced Actions Register (ASF Controller—B1:D8:F0)

Offset Address: F3h Attribute: R/W
Default Value: 02h Size: 8 bits

This register contains many different forcible actions including APM functions, flushing internal pending SOS operations, software SOS operations, software reset, and EEPROM reload. Writes to this register must only set one bit per-write. Setting multiple bits in a single write can have indeterminate results.

***Note:*** For bits in this register, writing a 1 invokes the operation. The bits self-clear immediately.

| Bit | Description |
|---|---|
| 7 | **Software Reset (FRC_RST) — R/W**. This bit is used to reset the ASF controller. It performs the equivalent of a hardware reset and re-read the EEPROM. This bit self-clears immediately. Software should wait for the EEC_LOAD bit to clear. |
| 6 | **Force EEPROM Reload (FRC_EELD) — R/W**. Force Reload of EEPROM without affect current monitoring state of the ASF controller. This bit self-clears immediately.<br>**NOTE:** Software registers in EEPROM are not loaded by this action. Software should disable the ASF controller before issuing this command and wait for STA_LOAD to clear before enabling again. |
| 5 | **Flush SOS (FRC_FLUSH) — R/W**. This bit is used to flush any pending SOSes or history internal to the ASF controller. This is necessary because the Status register only shows events that have happened as opposed to SOS events sent. Also, the history bits in the ASF controller are not software visible. Self-clears immediately. |
| 4 | **Reserved** |
| 3 | **Force APM Power Cycle (FRC_ACYC) — R/W**. This mode forces the ASF controller to initiate a power cycle to the system. The bit self-clears immediately. |
| 2 | **Force APM Hard Power Down (FRC_AHDN) — R/W**. This mode forces the ASF controller to initiate a hard power down of the system immediately. The bit self-clears immediately. |
| 1 | **Clear ASF Polling History (FRC_CLRAPOL) — R/W**. Writing a 1b to this bit position will clear the Poll History associated with all ASF Polling. Writing a 0b has no effect. This bit self-clears immediately. |
| 0 | **Force APM Reset (FRC_ARST) — R/W**. This mode forces the ASF controller to initiate a hard reset of the system immediately. The bit self-clears immediately. |

### 11.3.17 RMCP_SNUM—RMCP Sequence Number Register (ASF Controller—B1:D8:F0)

Offset Address: F4h
Default Value: 00h
Attribute: R/W
Size: 8 bits

This register is a means for software to read the current sequence number that hardware is using in RMCP packets. Software can also change the value. Software should only write to this register while the GLOBAL ENABLE is off.

| Bit | Description |
|---|---|
| 7:0 | **RMCP Sequence Num**ber (RSEQ_VAL) — R/W. This is the current sequence number of the RMCP packet being sent or the sequence number of the next RMCP packet to be sent. This value can be set by software. At reset, it defaults to 00h. If the sequence number is not FFh, the ASF controller will automatically increment this number by one (or rollover to 00h if incrementing from FEh) after a successful RMCP packet transmission. |

### 11.3.18 SP_MODE—Special Modes Register (ASF Controller—B1:D8:F0)

Offset Address: F5h
Default Value: x0h
Attribute: R/WC, RO
Size: 8 bits

The register contains miscellaneous functions.

| Bit | Description |
|---|---|
| 7 | **SMBus Activity Bit (SPE_ACT) — RO**.<br>1 = ASF controller is active with a SMBus transaction. This is an indicator to software that the ASF controller is still processing commands on the SMBus. |
| 6 | **Watchdog Status (SPE_WDG) — R/WC**.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit is set when a watchdog expiration occurs. |
| 5 | **Link Loss Status (SPE_LNK) — R/WC**.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit is set when a link loss occurs (link is down for more than 5 seconds). |
| 4:0 | **Reserved** |

### 11.3.19 INPOLL_TCONF—Inter-Poll Timer Configuration Register (ASF Controller—B1:D8:F0)

Offset Address: F6h
Default Value: 10h
Attribute: R/W
Size: 8 bits

This register is used to load and hold the value (in increments of 5 ms) for the polling timer. This value determines how often the ASF polling timer expires which determines the minimum idle time between sensor polls.

| Bit | Description |
|---|---|
| 7:0 | **Inter-Poll Timer Configuration (IPTC_VAL) — R/W**. This field identifies the time, in 5.24 ms units that the ASF controller will wait between the end of the one ASF Poll Alert Message to start on the next. The value 00h is invalid and unsupported. |

## 11.3.20 PHIST_CLR—Poll History Clear Register (ASF Controller—B1:D8:F0)

Offset Address: F7h
Default Value: 00h
Attribute: R/WC
Size: 8 bits

This register is used to clear the history of the Legacy Poll operations. ASF maintains history of the last poll data for each Legacy Poll operation to compare against the current poll to detect changes. By setting the appropriate bit, the history for that Legacy Poll is cleared to 0s.

| Bit | Description |
|---|---|
| 7 | **Clear Polling Descriptor 8 History (PHC_POLL8) — R/WC**. Writing a 1b to this bit position will clear the Poll History associated with Polling Descriptor #8. Writing a 0b has no effect. |
| 6 | **Clear Polling Descriptor 7 History (PHC_POLL7) — R/WC**. Writing a 1b to this bit position will clear the Poll History associated with Polling Descriptor #7. Writing a 0b has no effect. |
| 5 | **Clear Polling Descriptor 6 History (PHC_POLL6) — R/WC**. Writing a 1b to this bit position will clear the Poll History associated with Polling Descriptor #6. Writing a 0b has no effect. |
| 4 | **Clear Polling Descriptor 5 History (PHC_POLL5) — R/WC**. Writing a 1b to this bit position will clear the Poll History associated with Polling Descriptor #5. Writing a 0b has no effect. |
| 3 | **Clear Polling Descriptor 4 History (PHC_POLL4) — R/WC**. Writing a 1b to this bit position will clear the Poll History associated with Polling Descriptor #4. Writing a 0b has no effect. |
| 2 | **Clear Polling Descriptor 3 History (PHC_POLL3) — R/WC**. Writing a 1b to this bit position will clear the Poll History associated with Polling Descriptor #3. Writing a 0b has no effect. |
| 1 | **Clear Polling Descriptor 2 History (PHC_POLL2) — R/WC**. Writing a 1b to this bit position will clear the Poll History associated with Polling Descriptor #2. Writing a 0b has no effect. |
| 0 | **Clear Polling Descriptor 1 History (PHC_POLL1) — R/WC**. Writing a 1b to this bit position will clear the Poll History associated with Polling Descriptor #1. Writing a 0b has no effect. |

### 11.3.21 PMSK1—Polling Mask 1 Register (ASF Controller—B1:D8:F0)

Offset Address: F8h
Default Value: XXh
Attribute: R/W
Size: 8 bits

This register provides software an interface for the Polling #1 Data Mask.

| Bit | Description |
|---|---|
| 7:0 | **Polling Mask for Polling Descriptor #1 (POL1_MSK) — R/W**. This field is used to read and write the data mask for Polling Descriptor #1. Software should only access this register when the ASF controller is GLOBAL DISABLED. |

### 11.3.22 PMSK2—Polling Mask 2 Register (ASF Controller—B1:D8:F0)

Offset Address: F9h
Default Value: XXh
Attribute: R/W
Size: 8 bits

This register provides software an interface for the Polling #2 Data Mask.

| Bit | Description |
|---|---|
| 7:0 | **Polling Mask for Polling Descriptor #2 (POL2_MSK)** — R/W. This field is used to read and write the data mask for Polling Descriptor #2. Software should only access this register when the ASF controller is GLOBAL DISABLED. |

### 11.3.23 PMSK3—Polling Mask 3 Register (ASF Controller—B1:D8:F0)

Offset Address: FAh
Default Value: XXh
Attribute: R/W
Size: 8 bits

This register provides software an interface for the Polling #3 Data Mask.

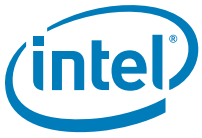
| Bit | Description |
|---|---|
| 7:0 | **Polling Mask for Polling Descriptor #3 (POL3_MSK) — R/W**. This register is used to read and write the data mask for Polling Descriptor #3. Software should only access this register when the ASF controller is GLOBAL DISABLED. |

### 11.3.24 PMSK4—Polling Mask 4 Register (ASF Controller—B1:D8:F0)

Offset Address: FBh
Default Value: XXh
Attribute: R/W
Size: 8 bits

This register provides software an interface for the Polling #4 Data Mask.

| Bit | Description |
|---|---|
| 7:0 | **Polling Mask for Polling Descriptor #4 (POL4_MSK) — R/W**. This register is used to read and write the data mask for Polling Descriptor #4. Software should only access this register when the ASF controller is GLOBAL DISABLED. |



### 11.3.25 PMSK5—Polling Mask 5 Register (ASF Controller—B1:D8:F0)

Offset Address: FCh
Default Value: XXh
Attribute: R/W
Size: 8 bits

This register provides software an interface for the Polling #5 Data Mask.

| Bit | Description |
|---|---|
| 7:0 | **Polling Mask for Polling Descriptor #5 (POL5_MSK) — R/W**. This register is used to read and write the data mask for Polling Descriptor #5. Software should only access this register when the ASF controller is GLOBAL DISABLED. |

### 11.3.26 PMSK6—Polling Mask 6 Register (ASF Controller—B1:D8:F0)

Offset Address: FDh
Default Value: XXh
Attribute: R/W
Size: 8 bits

This register provides software an interface for the Polling #6 Data Mask.

| Bit | Description |
|---|---|
| 7:0 | **Polling Mask for Polling Descriptor #6 (POL6_MSK) — R/W**. This register is used to read and write the data mask for Polling Descriptor #6. Software should only access this register when the ASF controller is GLOBAL DISABLED. |

### 11.3.27 PMSK7—Polling Mask 7 Register (ASF Controller—B1:D8:F0)

Offset Address: FEh
Default Value: XXh
Attribute: R/W
Size: 8 bits

This register provides software an interface for the Polling #7 Data Mask.

| Bit | Description |
|---|---|
| 7:0 | **Polling Mask for Polling Descriptor #7 (POL7_MSK) — R/W**. This register is used to read and write the data mask for Polling Descriptor #7. Software should only access this register when the ASF controller is GLOBAL DISABLED. |

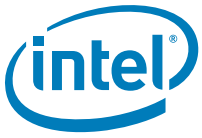
### 11.3.28 PMSK8—Polling Mask 8 Register (ASF Controller—B1:D8:F0)

Offset Address: FFh
Default Value: XXh
Attribute: R/W
Size: 8 bits

This register provides software an interface for the Polling #8 Data Mask.

| Bit | Description |
|---|---|
| 7:0 | **Polling Mask for Polling Descriptor #8 (POL8_MSK) — R/W.** This register is used to read and write the data mask for Polling Descriptor #8. Software should only access this register when the ASF controller is GLOBAL DISABLED. |

§



# 12 PCI-to-PCI Bridge Registers (D30:F0)

The Chipset PCI bridge resides in PCI Device 30, Function 0 on bus #0. This implements the buffering and control logic between PCI and the backbone. The arbitration for the PCI bus is handled by this PCI device.

## 12.1 PCI Configuration Registers (D30:F0)

***Note:*** Address locations that are not shown should be treated as **Reserved** (see Section 9.2 for details).

**Table 12-118.PCI Bridge Register Address Map (PCI-PCI—D30:F0) (Sheet 1 of 2)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00h–01h | VID | Vendor Identification | 8086h | RO |
| 02h–03h | DID | Device Identification | See register description | RO |
| 04h–05h | PCICMD | PCI Command | 0000h | R/W, RO |
| 06h–07h | PSTS | PCI Status | 0010h | R/WC, RO |
| 08h | RID | Revision Identification | See register description | RO |
| 09h-0Bh | CC | Class Code | 00060401h | RO |
| 0Dh | PMLT | Primary Master Latency Timer | 00h | RO |
| 0Eh | HEADTYP | Header Type | 81h | RO |
| 18h-1Ah | BNUM | Bus Number | 000000h | R/W, RO |
| 1Bh | SMLT | Secondary Master Latency Timer | 00h | R/W, RO |
| 1Ch-1Dh | IOBASE_LIMIT | I/O Base and Limit | 0000h | R/W, RO |
| 1Eh–1Fh | SECSTS | Secondary Status | 0280h | R/WC, RO |
| 20h–23h | MEMBASE_LIMIT | Memory Base and Limit | 00000000h | R/W, RO |
| 24h–27h | PREF_MEM_BASE_LIMIT | Prefetchable Memory Base and Limit | 00010001h | R/W, RO |
| 28h–2Bh | PMBU32 | Prefetchable Memory Upper 32 Bits | 00000000h | R/W |
| 2Ch–2Fh | PMLU32 | Prefetchable Memory Limit Upper 32 Bits | 00000000h | R/W |
| 34h | CAPP | Capability List Pointer | 50h | RO |
| 3Ch-3Dh | INTR | Interrupt Information | 0000h | R/W, RO |
| 3Eh–3Fh | BCTRL | Bridge Control | 0000h | R/WC, RO |
| 40h–41h | SPDH | Secondary PCI Device Hiding | 00h | R/W, RO |
| 44h-47h | DTC | Delayed Transaction Control | 00000000h | R/W, RO |
| 48h-4Bh | BPS | Bridge Proprietary Status | 00000000h | R/WC, RO |

**Table 12-118.PCI Bridge Register Address Map (PCI-PCI—D30:F0) (Sheet 2 of 2)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 4Ch-4Fh | BPC | Bridge Policy Configuration | 00001200h | R/W RO |
| 50–51h | SVCAP | Subsystem Vendor Capability Pointer | 000Dh | RO |
| 54h-57h | SVID | Subsystem Vendor IDs | 00000000 | R/WO |

## 12.1.1 VID— Vendor Identification Register (PCI-PCI—D30:F0)

Offset Address: 00h–01h
Default Value: 8086h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Vendor ID — RO.** This is a 16-bit value assigned to Intel. Intel VID = 8086h. |

## 12.1.2 DID— Device Identification Register (PCI-PCI—D30:F0)

Offset Address: 02h–03h
Default Value: See bit description
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Device ID — RO**.This is a 16-bit value assigned to the PCI bridge. Refer to the Intel® I/O Controller Hub 7 Family Specification Update for the value of the Device ID Register. |

## 12.1.3 PCICMD—PCI Command (PCI-PCI—D30:F0)

Offset Address: 04h–05h
Default Value: 0000h
Attribute: R/W, RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:11 | **Reserved** |
| 10 | **Interrupt Disable (ID) — RO**. Hardwired to 0. The PCI bridge has no interrupts to disable |
| 9 | Fast Back to Back Enable (FBE) — RO. Hardwired to 0, per the *PCI Express* Base Specification, Revision 1.0a*. |
| 8 | **SERR# Enable (SERR_EN) — R/W**.<br>0 = Disable.<br>1 = Enable the Chipset to generate an NMI (or SMI# if NMI routed to SMI#) when the D30:F0 SSE bit (offset 06h, bit 14) is set. |
| 7 | Wait Cycle Control (WCC) — RO. Hardwired to 0, per the *PCI Express* Base Specification, Revision 1.0a*. |
| 6 | **Parity Error Response (PER) — R/W**.<br>0 = The Chipset ignores parity errors on the PCI bridge.<br>1 = The Chipset will set the SSE bit (D30:F0, offset 06h, bit 14) when parity errors are detected on the PCI bridge. |
| 5 | **VGA Palette Snoop (VPS) — RO**. Hardwired to 0, per the *PCI Express* Base Specification, Revision 1.0a*. |

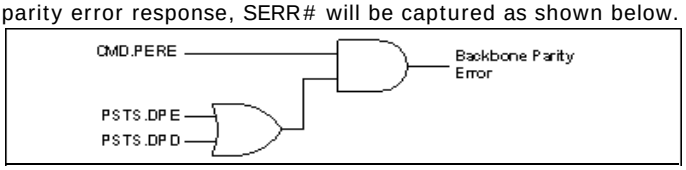
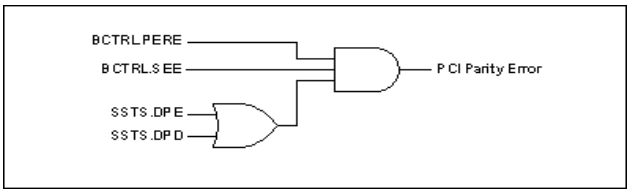
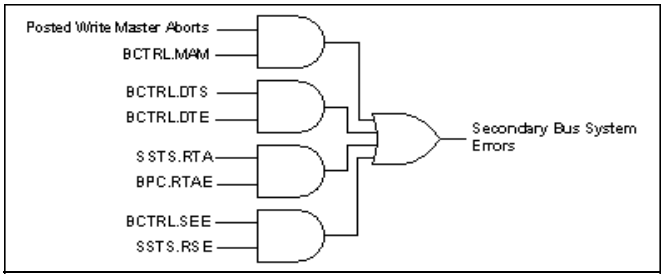
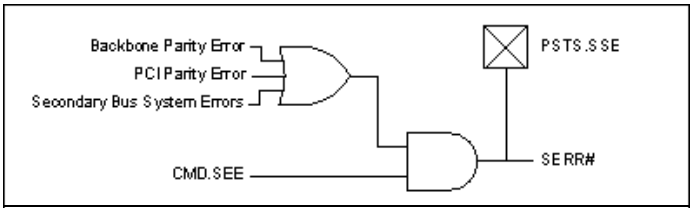
| Bit | Description |
|---|---|
| 4 | **Memory Write and Invalidate Enable (MWE) — RO**. Hardwired to 0, per the *PCI Express* Base Specification, Revision 1.0a* |
| 3 | **Special Cycle Enable (SCE) — RO**. Hardwired to 0, per the *PCI Express* Base Specification, Revision 1.0a* and the *PCI- to-PCI Bridge Specification.* |
| 2 | **Bus Master Enable (BME)** — R/W.<br>0 = Disable<br>1 = Enable. Allows the PCI-to-PCI bridge to accept cycles from PCI. |
| 1 | **Memory Space Enable (MSE)** — R/W. Controls the response as a target for memory cycles targeting PCI.<br>0 = Disable<br>1 = Enable |
| 0 | **I/O Space Enable (IOSE)** — R/W. Controls the response as a target for I/O cycles targeting PCI.<br>0 = Disable<br>1 = Enable |

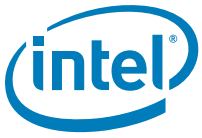
## 12.1.4 PSTS—PCI Status Register (PCI-PCI—D30:F0)

Offset Address: 06h–07h Attribute: R/WC, RO
Default Value: 0010h Size: 16 bits

***Note:*** For the writable bits, software must write a 1 to clear bits that are set. Writing a 0 to the bit has no effect.

| Bit | Description |
|---|---|
| 15 | **Detected Parity Error (DPE)** — R/WC.<br>0 = Parity error Not detected.<br>1 = Indicates that the Chipset detected a parity error on the internal backbone. This bit gets set even if the Parity Error Response bit (D30:F0:04 bit 6) is not set. |

| Bit | Description |
|---|---|
| 14 | **Signaled System Error (SSE)** — R/WC. Several internal and external sources of the bridge can cause SERR#. The first class of errors is parity errors related to the backbone. The PCI bridge captures generic data parity errors (errors it finds on the backbone) as well as errors returned on backbone cycles where the bridge was the master. If either of these two conditions is met, and the primary side of the bridge is enabled for parity error response, SERR# will be captured as shown below.<br>  <br>As with the backbone, the PCI bus captures the same sets of errors. The PCI bridge captures generic data parity errors (errors it finds on PCI) as well as errors returned on PCI cycles where the bridge was the master. If either of these two conditions is met, and the secondary side of the bridge is enabled for parity error response, SERR# will be captured as shown below.<br>  <br>The final class of errors is system bus errors. There are three status bits associated with system bus errors, each with a corresponding enable. The diagram capturing this is shown below.<br>  <br>After checking for the three above classes of errors, an SERR# is generated, and PSTS.SSE logs the generation of SERR#, if CMD.SEE (D30:F0:04, bit 8) is set, as shown below.<br>  |
| 13 | **Received Master Abort (RMA) — R/WC.**<br>0 = No master abort received.<br>1 = Set when the bridge receives a master abort status from the backbone. |
| 12 | **Received Target Abort (RTA) — R/WC.**<br>0 = No target abort received.<br>1 = Set when the bridge receives a target abort status from the backbone. |



| Bit | Description |
|---|---|
| 11 | **Signaled Target Abort (STA) — R/WC**.<br>0 = No signaled target abort<br>1 = Set when the bridge generates a completion packet with target abort status on the backbone. |
| 10:9 | **Reserved**. |
| 8 | **Data Parity Error Detected (DPD) — R/WC**.<br>0 = Data parity error Not detected.<br>1 = Set when the bridge receives a completion packet from the backbone from a previous request, and detects a parity error, and CMD.PERE is set (D30:F0:04 bit 6). |
| 7:5 | **Reserved**. |
| 4 | **Capabilities List (CLIST) — RO.** Hardwired to 1. Capability list exist on the PCI bridge. |
| 3 | **Interrupt Status (IS) — RO**. Hardwired to 0. The PCI bridge does not generate interrupts. |
| 2:0 | **Reserved** |

## 12.1.5 RID—Revision Identification Register (PCI-PCI—D30:F0)

Offset Address: 08h Attribute: RO
Default Value: See bit description Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Revision ID — RO** |

## 12.1.6 CC—Class Code Register (PCI-PCI—D30:F0)

Offset Address: 09h-0Bh Attribute: RO
Default Value: 060401h Size: 24 bits

| Bit | Description |
|---|---|
| 23:16 | **Base Class Code (BCC) — RO**. Hardwired to 06h. Indicates this is a bridge device. |
| 15:8 | **Sub Class Code (SCC) — RO**. Hardwired to 04h. Indicates this device is a PCI-to-PCI bridge. |
| 7:0 | **Programming Interface (PI) — RO**. Hardwired to 01h. Indicates the bridge is subtractive decode |

### 12.1.7 PMLT—Primary Master Latency Timer Register (PCI-PCI—D30:F0)

Offset Address: 0Dh Attribute: RO
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:3 | **Master Latency Timer Count (MLTC) — RO**. **Reserved** per the *PCI Express* Base Specification, Revision 1.0a*. |
| 2:0 | **Reserved** |

### 12.1.8 HEADTYP—Header Type Register (PCI-PCI—D30:F0)

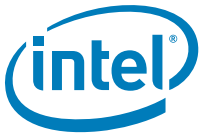
Offset Address: 0Eh Attribute: RO
Default Value: 81h Size: 8 bits

| Bit | Description |
|---|---|
| 7 | **Multi-Function Device (MFD) — RO**. The value reported here depends upon the state of the AC '97 function hide (FD) register (Chipset Config Registers:Offset 3418h), per the following table:<br><br>FD.AAD / FD.AMD / MFD<br>0 / 0 / 1<br>0 / 1 / 1<br>1 / 0 / 1<br>1 / 1 / 0 |
| 6:0 | Header Type (HTYPE) — RO. This 7-bit field identifies the header layout of the configuration space, which is a PCI-to-PCI bridge in this case. |

### 12.1.9 BNUM—Bus Number Register (PCI-PCI—D30:F0)

Offset Address: 18h-1Ah Attribute: R/W, RO
Default Value: 000000h Size: 24 bits

| Bit | Description |
|---|---|
| 23:16 | **Subordinate Bus Number (SBBN) — R/W**. Indicates the highest PCI bus number below the bridge. |
| 15:8 | **Secondary Bus Number (SCBN) — R/W.** Indicates the bus number of PCI. |
| 7:0 | **Primary Bus Number (PBN) — RO**. Hardwired to 00h for legacy software compatibility. |



### 12.1.10 SMLT—Secondary Master Latency Timer Register (PCI-PCI—D30:F0)

Offset Address: 1Bh
Default Value: 00h
Attribute: R/W, RO
Size: 8 bits

This timer controls the amount of time the Chipset PCI-to-PCI bridge will burst data on its secondary interface. The counter starts counting down from the assertion of FRAME#. If the grant is removed, then the expiration of this counter will result in the de-assertion of FRAME#. If the grant has not been removed, then the Chipset PCI-to-PCI bridge may continue ownership of the bus.

| Bit | Description |
|---|---|
| 7:3 | **Master Latency Timer Count (MLTC) — R/W**. This 5-bit field indicates the number of PCI clocks, in 8-clock increments, that the Chipset remains as master of the bus. |
| 2:0 | **Reserved** |

### 12.1.11 IOBASE_LIMIT—I/O Base and Limit Register (PCI-PCI—D30:F0)

Offset Address: 1Ch-1Dh
Default Value: 0000h
Attribute: R/W, RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:12 | **I/O Limit Address Limit bits[15:12] — R/W**. I/O Base bits corresponding to address lines 15:12 for 4-KB alignment. Bits 11:0 are assumed to be padded to FFFh. |
| 11:8 | **II/O Limit Address Capability (IOLC) — RO**. Indicates that the bridge does not support 32-bit I/O addressing. |
| 7:4 | **I/O Base Address (IOBA) — R/W**. I/O Base bits corresponding to address lines 15:12 for 4-KB alignment. Bits 11:0 are assumed to be padded to 000h. |
| 3:0 | **I/O Base Address Capability (IOBC) — RO**. Indicates that the bridge does not support 32-bit I/O addressing. |

### 12.1.12 SECSTS—Secondary Status Register (PCI-PCI—D30:F0)

Offset Address: 1Eh–1Fh
Default Value: 0280h
Attribute: R/WC, RO
Size: 16 bits

***Note:*** For the writable bits, software must write a 1 to clear bits that are set. Writing a 0 to the bit has no effect.

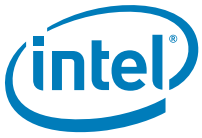
| Bit | Description |
|---|---|
| 15 | **Detected Parity Error (DPE) — R/WC.**<br>0 = Parity error **not** detected.<br>1 = Chipset PCI bridge detected an address or data parity error on the PCI bus |
| 14 | **Received System Error (RSE) — R/WC.**<br>0 = SERR# assertion **not** received<br>1 = SERR# assertion is received on PCI. |

| Bit | Description |
|---|---|
| 13 | **Received Master Abort (RMA) — R/WC.**<br>0 = No master abort.<br>1 = This bit is set whenever the bridge is acting as an initiator on the PCI bus and the cycle is master-aborted. For (G)MCH or CPU/Chipset interface packets that have completion required, this must also cause a target abort to be returned and sets PSTS.STA. (D30:F0:06 bit 11) |
| 12 | **Received Target Abort (RTA) — R/WC.**<br>0 = No target abort.<br>1 = This bit is set whenever the bridge is acting as an initiator on PCI and a cycle is target-aborted on PCI. For (G)MCH or CPU/Chipset interface packets that have completion required, this event must also cause a target abort to be returned, and sets PSTS.STA. (D30:F0:06 bit 11). |
| 11 | **Signaled Target Abort (STA) — R/WC.**<br>0 = No target abort.<br>1 = This bit is set when the bridge is acting as a target on the PCI Bus and signals a target abort. |
| 10:9 | **DEVSEL# Timing (DEVT) — RO.**<br>01h = Medium decode timing. |
| 8 | **Data Parity Error Detected (DPD) — R/WC.**<br>0 = Conditions described below **not** met.<br>1 = The Chipset sets this bit when all of the following three conditions are met:<br>• The bridge is the initiator on PCI.<br>• PERR# is detected asserted or a parity error is detected internally<br>• BCTRL.PERE (D30:F0:3E bit 0) is set. |
| 7 | **Fast Back to Back Capable (FBC) — RO**. Hardwired to 1 to indicate that the PCI to PCI target logic is capable of receiving fast back-to-back cycles. |
| 6 | **Reserved** |
| 5 | **66 MHz Capable (66MHZ_CAP) — RO**. Hardwired to 0. This bridge is 33 MHz capable only. |
| 4:0 | **Reserved** |

## 12.1.13 MEMBASE_LIMIT—Memory Base and Limit Register (PCI-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 20h–23h | Attribute: | R/W, RO |
| Default Value: | 00000000h | Size: | 32 bits |

This register defines the base and limit, aligned to a 1-MB boundary, of the non-prefetchable memory area of the bridge. Accesses that are within the ranges specified in this register will be sent to PCI if CMD.MSE is set. Accesses from PCI that are outside the ranges specified will be accepted by the bridge if CMD.BME is set.



| Bit | Description |
|---|---|
| 31-20 | **Memory Limit (ML) — R/W**. These bits are compared with bits 31:20 of the incoming address to determine the upper 1-MB aligned value (exclusive) of the range. The incoming address must be less than this value. |
| 19-16 | **Reserved** |
| 15:4 | **Memory Base (MB) — R/W**. These bits are compared with bits 31:20 of the incoming address to determine the lower 1-MB aligned value (inclusive) of the range. The incoming address must be greater than or equal to this value. |
| 3:0 | **Reserved** |

## 12.1.14 PREF_MEM_BASE_LIMIT—Prefetchable Memory Base and Limit Register (PCI-PCI—D30:F0)

Offset Address: 24h–27h Attribute: R/W, RO
Default Value: 00010001h Size: 32-bit

Defines the base and limit, aligned to a 1-MB boundary, of the prefetchable memory area of the bridge. Accesses that are within the ranges specified in this register will be sent to PCI if CMD.MSE is set. Accesses from PCI that are outside the ranges specified will be accepted by the bridge if CMD.BME is set.

| Bit | Description |
|---|---|
| 31-20 | **Prefetchable Memory Limit (PML)** — R/W. These bits are compared with bits 31:20 of the incoming address to determine the upper 1-MB aligned value (exclusive) of the range. The incoming address must be less than this value. |
| 19-16 | 64-bit Indicator (I64L) — RO. Indicates support for 64-bit addressing. |
| 15:4 | **Prefetchable Memory Base (PMB)** — R/W. These bits are compared with bits 31:20 of the incoming address to determine the lower 1-MB aligned value (inclusive) of the range. The incoming address must be greater than or equal to this value. |
| 3:0 | 64-bit Indicator (I64B) — RO. Indicates support for 64-bit addressing. |

## 12.1.15 PMBU32—Prefetchable Memory Base Upper 32 Bits Register (PCI-PCI—D30:F0)

Offset Address: 28h–2Bh Attribute: R/W
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Prefetchable Memory Base Upper Portion (PMBU)** — R/W. Upper 32-bits of the prefetchable address base. |

### 12.1.16 PMLU32—Prefetchable Memory Limit Upper 32 Bits Register (PCI-PCI—D30:F0)

Offset Address: 2C–2Fh Attribute: R/W
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Prefetchable Memory Limit Upper Portion (PMLU)** — R/W. Upper 32-bits of the prefetchable address limit. |

### 12.1.17 CAPP—Capability List Pointer Register (PCI-PCI—D30:F0)

Offset Address: 34h Attribute: RO
Default Value: 50h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Capabilities Pointer (PTR) — RO.** Indicates that the pointer for the first entry in the capabilities list is at 50h in configuration space. |

### 12.1.18 INTR—Interrupt Information Register (PCI-PCI—D30:F0)

Offset Address: 3Ch–3Dh Attribute: R/W, RO
Default Value: 0000h Size: 16 bits

| Bit | Description |
|---|---|
| 15:8 | **Interrupt Pin (IPIN) — RO**. The PCI bridge does not assert an interrupt. |
| 7:0 | **Interrupt Line (ILINE)** — R/W. Software written value to indicate which interrupt line (vector) the interrupt is connected to. No hardware action is taken on this register. Since the bridge does not generate an interrupt, BIOS should program this value to FFh as per the PCI bridge specification. |

### 12.1.19 BCTRL—Bridge Control Register (PCI-PCI—D30:F0)

Offset Address: 3Eh–3Fh Attribute: R/WC, RO
Default Value: 0000h Size: 16 bits

| Bit | Description |
|---|---|
| 15:12 | **Reserved** |
| 11 | **Discard Timer SERR# Enable (DTE) — R/W**. Controls the generation of SERR# on the primary interface in response to the DTS bit being set:<br>0 = Do not generate SERR# on a secondary timer discard<br>1 = Generate SERR# in response to a secondary timer discard |
| 10 | **Discard Timer Status (DTS) — R/WC**. This bit is set to 1 when the secondary discard timer (see the SDT bit below) expires for a delayed transaction in the hard state. |

| Bit | Description |
|---|---|
| 9 | **Secondary Discard Timer (SDT) — R/W.** This bit sets the maximum number of PCI clock cycles that the Chipset waits for an initiator on PCI to repeat a delayed transaction request. The counter starts once the delayed transaction data is has been returned by the system and is in a buffer in the Chipset PCI bridge. If the master has not repeated the transaction at least once before the counter expires, the Chipset PCI bridge discards the transaction from its queue.<br>0 = The PCI master timeout value is between $2^{15}$ and $2^{16}$ PCI clocks<br>1 = The PCI master timeout value is between $2^{10}$ and $2^{11}$ PCI clocks |
| 8 | **Primary Discard Timer (PDT) — R/W**. This bit is R/W for software compatibility only. |
| 7 | Fast Back to Back Enable (FBE) — RO. Hardwired to 0. The PCI logic will not generate fast back-to-back cycles on the PCI bus. |
| 6 | **Secondary Bus Reset (SBR) — R/W**. This bit controls PCIRST# assertion on PCI.<br>0 = Bridge de-asserts PCIRST#<br>1 = Bridge asserts PCIRST#. When PCIRST# is asserted, the delayed transaction buffers, posting buffers, and the PCI bus are initialized back to reset conditions. The rest of the part and the configuration registers are not affected. |
| 5 | **Master Abort Mode (MAM) — R/W**. This bit controls the Chipset PCI bridge's behavior when a master abort occurs:<br>Master Abort on (G)MCH or CPU/Chipset Interconnect (DMI):<br>0 = Bridge asserts TRDY# on PCI. It drives all 1's for reads, and discards data on writes.<br>1 = Bridge returns a target abort on PCI.<br>Master Abort PCI (non-locked cycles):<br>0 = Normal completion status will be returned on the (G)MCH or CPU/Chipset interconnect.<br>1 = Target abort completion status will be returned on the (G)MCH or CPU/Chipset interconnect.<br>**NOTE:** All locked reads will return a completer abort completion status on the (G)MCH or CPU/Chipset interconnect. |
| 4 | **VGA 16-Bit Decode (V16D) — R/W**. This bit controls enables the Chipset PCI bridge to provide 16-bits decoding of VGA I/O address precluding the decode of VGA alias addresses every 1 KB. This bit requires the VGAE bit in this register be set. |
| 3 | **VGA Enable (VGAE) — R/W**. When set to a 1, the Chipset PCI bridge forwards the following transactions to PCI regardless of the value of the I/O base and limit registers. The transactions are qualified by CMD.MSE (D30:F0:04 bit 1) and CMD.IOSE (D30:F0:04 bit 0) being set.<br>• Memory addresses: 000A0000h-000BFFFFh<br>• I/O addresses: 3B0h-3BBh and 3C0h-3DFh. For the I/O addresses, bits [63:16] of the address must be 0, and bits [15:10] of the address are ignored (i.e., aliased).<br>The same holds true from secondary accesses to the primary interface in reverse. That is, when the bit is 0, memory and I/O addresses on the secondary interface between the above ranges will be claimed. |

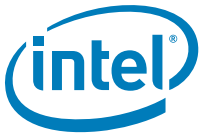
| Bit | Description |
|---|---|
| 2 | **ISA Enable (IE) — R/W**. This bit only applies to I/O addresses that are enabled by the I/O Base and I/O Limit registers and are in the first 64 KB of PCI I/O space. If this bit is set, the Chipset PCI bridge will block any forwarding from primary to secondary of I/O transactions addressing the last 768 bytes in each 1-KB block (offsets 100h to 3FFh). |
| 1 | **SERR# Enable (SEE) — R/W**. This bit controls the forwarding of secondary interface SERR# assertions on the primary interface. When set, the PCI bridge will forward SERR# pin.<br>• SERR# is asserted on the secondary interface.<br>• This bit is set.<br>• CMD.SEE (D30:F0:04 bit 8) is set. |
| 0 | **Parity Error Response Enable (PERE) — R/W**.<br>0 = Disable<br>1 = The Chipset PCI bridge is enabled for parity error reporting based on parity errors on the PCI bus. |

## 12.1.20 SPDH—Secondary PCI Device Hiding Register (PCI-PCI—D30:F0)

Offset Address: 40h–41h
Default Value: 00h
Attribute: R/W, RO
Size: 16 bits

This register allows software to hide the PCI devices, either plugged into slots or on the motherboard.

| Bit | Description |
|---|---|
| 15:8 | **Reserved** |
| 7 | **Hide Device 7 (HD7) — R/W, RO**. Same as bit 0 of this register, except for device 7 (AD[23]) |
| 6 | **Hide Device 6 (HD6) — R/W, RO**. Same as bit 0 of this register, except for device 6 (AD[22]) |
| 5 | **Hide Device 5 (HD5) — R/W, RO**. Same as bit 0 of this register, except for device 5 (AD[21]) |
| 4 | **Hide Device 4 (HD4) — R/W, RO**. Same as bit 0 of this register, except for device 4 (AD[20]) |
| 3 | **Hide Device 3 (HD3) — R/W, RO**. Same as bit 0 of this register, except for device 3 (AD[19]) |
| 2 | **Hide Device 2 (HD2) — R/W, RO.** Same as bit 0 of this register, except for device 2 (AD[18]) |
| 1 | **Hide Device 1 (HD1) — R/W, RO**. Same as bit 0 of this register, except for device 1 (AD[17]) |
| 0 | **Hide Device 0 (HD0) — R/W, RO**.<br>0 = The PCI configuration cycles for this slot are not affected.<br>1 = Chipset hides device 0 on the PCI bus. This is done by masking the IDSEL (keeping it low) for configuration cycles to that device. Since the device will not see its IDSEL go active, it will not respond to PCI configuration cycles and the processor will think the device is not present. AD[16] is used as IDSEL for device 0. |



## 12.1.21 DTC—Delayed Transaction Control Register (PCI-PCI—D30:F0)

Offset Address: 44h–47h Attribute: R/W, RO
Default Value: 00000000h Size: 32 bits

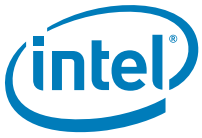
| Bit | Description |
|---|---|
| 31 | **Discard Delayed Transactions (DDT) — R/W.**<br>0 = Logged delayed transactions are kept.<br>1 = The Chipset PCI bridge will discard any delayed transactions it has logged. This includes transactions in the pending queue, and any transactions in the active queue, whether in the hard or soft DT state. The prefetchers will be disabled and return to an idle state.<br>**NOTE:** If a transaction is running on PCI at the time this bit is set, that transaction will continue until either the PCI master disconnects (by de-asserting FRAME#) or the PCI bridge disconnects (by asserting STOP#). This bit is cleared by the PCI bridge when the delayed transaction queues are empty and have returned to an idle state. Software sets this bit and polls for its completion |
| 30 | **Block Delayed Transactions (BDT) — R/W.**<br>0 = Delayed transactions accepted<br>1 = The Chipset PCI bridge will not accept incoming transactions which will result in delayed transactions. It will blindly retry these cycles by asserting STOP#. All postable cycles (memory writes) will still be accepted. |
| 29: 8 | **Reserved** |
| 7: 6 | **Maximum Delayed Transactions (MDT) — R/W**. This field controls the maximum number of delayed transactions that the Chipset PCI bridge will run. Encodings are:<br>00 =) 2 Active, 5 pending<br>01 =) 2 active, no pending<br>10 =) 1 active, no pending<br>11 =) **Reserved** |
| 5 | **Reserved** |
| 4 | **Auto Flush After Disconnect Enable (AFADE) — R/W.**<br>0 = The PCI bridge will retain any fetched data until required to discard by producer/ consumer rules.<br>1 = The PCI bridge will flush any prefetched data after either the PCI master (by de-asserting FRAME#) or the PCI bridge (by asserting STOP#) disconnects the PCI transfer. |
| 3 | **Never Prefetch (NP) — R/W.**<br>0 = Prefetch enabled<br>1 = The Chipset will only fetch a single DW and will not enable prefetching, regardless of the command being an Memory read (MR), Memory read line (MRL), or Memory read multiple (MRM). |

| Bit | Description |
|---|---|
| 2 | **Memory Read Multiple Prefetch Disable (MRMPD) — R/W.**<br>0 = MRM commands will fetch multiple cache lines as defined by the prefetch algorithm.<br>1 = Memory read multiple (MRM) commands will fetch only up to a single, 64-byte aligned cache line. |
| 1 | **Memory Read Line Prefetch Disable (MRLPD) — R/W.**<br>0 = MRL commands will fetch multiple cache lines as defined by the prefetch algorithm.<br>1 = Memory read line (MRL) commands will fetch only up to a single, 64-byte aligned cache line. |
| 0 | **Memory Read Prefetch Disable (MRPD) — R/W.**<br>0 = MR commands will fetch up to a 64-byte aligned cache line.<br>1 = Memory read (MR) commands will fetch only a single DW. |

### 12.1.22 BPS—Bridge Proprietary Status Register (PCI-PCI—D30:F0)

Offset Address: 48h–4Bh Attribute: R/WC, RO
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:17 | **Reserved** |
| 16 | **PERR# Assertion Detected (PAD) — R/WC.** This bit is set by hardware whenever the PERR# pin is asserted on the rising edge of PCI clock. This includes cases in which the chipset is the agent driving PERR#. It remains asserted until cleared by software writing a 1 to this location. When enabled by the PERR#-to-SERR# Enable bit (in the Bridge Policy Configuration register), a 1 in this bit can generate an internal SERR# and be a source for the NMI logic.<br>This bit can be used by software to determine the source of a system problem. |
| 15:7 | **Reserved** |
| 6:4 | **Number of Pending Transactions (NPT) — RO**. This field indicates to debug software how many transactions are in the pending queue. Possible values are:<br>000 = No pending transaction<br>001 = 1 pending transaction<br>010 = 2 pending transactions<br>011 = 3 pending transactions<br>100 = 4 pending transactions<br>101 = 5 pending transactions<br>110 - 111 = **Reserved**<br>**NOTE:** This field is not valid if DTC.MDT (offset 44h:bits 7:6) is any value other than '00'. |
| 3:2 | **Reserved** |
| 1:0 | **Number of Active Transactions (NAT) — RO.** This field indicates to debug software how many transactions are in the active queue. Possible values are:<br>00 = No active transactions<br>01 = 1 active transaction<br>10 = 2 active transactions<br>11 = **Reserved** |



## 12.1.23 BPC—Bridge Policy Configuration Register (PCI-PCI—D30:F0)

Offset Address: 4Ch–4Fh Attribute: R/W, RO
Default Value: 00001200h Size: 32 bits

| Bit | Description |
|---|---|
| 31:14 | **Reserved** |
| 13:8 | **Upstream Read Latency Threshold (URLT) — R/W**: This field specifies the number of PCI clocks after internally enqueuing an upstream memory read request at which point the PCI target logic should insert wait states in order to optimize lead-off latency. When the master returns after this threshold has been reached and data has not arrived in the Delayed Transaction completion queue, then the PCI target logic will insert wait states instead of immediately retrying the cycle. The PCI target logic will insert up to 16 clocks of target initial latency (from FRAME# assertion to TRDY# or STOP# assertion) before retrying the PCI read cycle (if the read data has not arrived yet).<br>Note that the starting event for this Read Latency Timer is not explicitly visible externally.<br>A value of 0h disables this policy completely such that wait states will not be inserted on the read lead-off data phase.<br>The default value (12h) specifies 18 PCI clocks (540 ns) and is approximately 4 clocks less than the typical idle lead-off latency expected for Nettop Chipset systems. This value may need to be changed by BIOS, depending on the platform. |
| 7 | **Subtractive Decode Policy (SDP) — R/W**.<br>0 = The PCI bridge always forwards memory and I/O cycles that are not claimed by any other device on the backbone (primary interface) to the PCI bus (secondary interface).<br>1 = The PCI bridge will not claim and forward memory or I/O cycles at all unless the corresponding Space Enable bit is set in the Command register.<br>**NOTE:** The Boot BIOS Destination Selection strap can force the BIOS accesses to PCI. |
| 6 | **PERR#-to-SERR# Enable (PSE) — R/W**. When this bit is set, a 1 in the PERR# Assertion status bit (in the Bridge Proprietary Status register) will result in an internal SERR# assertion on the primary side of the bridge (if also enabled by the SERR# Enable bit in the primary Command register). SERR# is a source of NMI. |
| 5 | **Secondary Discard Timer Testmode (SDTT) — R/W**.<br>0 = The secondary discard timer expiration will be defined in BCTRL.SDT (D30:F0:3E, bit 9)<br>1 = The secondary discard timer will expire after 128 PCI clocks. |
| 4:3 | **Reserved** |
| 2 | **Peer Decode Enable (PDE) — R/W**.<br>0 = The PCI bridge assumes that all memory cycles target main memory, and all I/O cycles are not claimed.<br>1 = The PCI bridge will perform peer decode on any memory or I/O cycle from PCI that falls outside of the memory and I/O window registers |
| 1 | **Reserved** |
| 0 | **Received Target Abort SERR# Enable (RTAE) — R/W.** When set, the PCI bridge will report SERR# when PSTS.RTA (D30:F0:06 bit 12) or SSTS.RTA (D30:F0:1E bit 12) are set, and CMD.SEE (D30:F0:04 bit 8) is set. |

### 12.1.24 SVCAP—Subsystem Vendor Capability Register (PCI-PCI—D30:F0)

Offset Address: 50h–51h Attribute: RO
Default Value: 000Dh Size: 16 bits

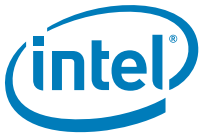
| Bit | Description |
|---|---|
| 15:8 | **Next Capability (NEXT) — RO**. Value of 00h indicates this is the last item in the list. |
| 7:0 | **Capability Identifier (CID) — RO**. Value of 0Dh indicates this is a PCI bridge subsystem vendor capability. |

### 12.1.25 SVID—Subsystem Vendor IDs Register (PCI-PCI—D30:F0)

Offset Address: 54h–57h Attribute: R/WO
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:16 | **Subsystem Identifier (SID) — R/WO**. This field indicates the subsystem as identified by the vendor. This field is write once and is locked down until a bridge reset occurs (not the PCI bus reset). |
| 15:0 | **Subsystem Vendor Identifier (SVID) — R/WO**. This field indicates the manufacturer of the subsystem. This field is write once and is locked down until a bridge reset occurs (not the PCI bus reset). |

§



# 13 LPC Interface Bridge Registers (D31:F0)

The LPC bridge function of the Chipset resides in PCI Device 31:Function 0. This function contains many other functional units, such as DMA and Interrupt controllers, Timers, Power Management, System Management, GPIO, RTC, and LPC Configuration Registers.

Registers and functions associated with other functional units (EHCI, UHCI, IDE, etc.) are described in their respective sections.

## 13.1 PCI Configuration Registers (LPC I/F—D31:F0)

***Note:*** Address locations that are not shown should be treated as Reserved.

**Table 13-119.LPC Interface PCI Register Address Map (LPC I/F—D31:F0) (Sheet 1 of 2)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00h–01h | VID | Vendor Identification | 8086h | RO |
| 02h–03h | DID | Device Identification | See register description. | RO |
| 04h–05h | PCICMD | PCI Command | 0007h | R/W, RO |
| 06h–07h | PCISTS | PCI Status | 0200h | R/WC, RO |
| 08h | RID | Revision Identification | See register description. | RO |
| 09h | PI | Programming Interface | 00h | RO |
| 0Ah | SCC | Sub Class Code | 01h | RO |
| 0Bh | BCC | Base Class Code | 06h | RO |
| 0Dh | PLT | Primary Latency Timer | 00h | RO |
| 0Eh | HEADTYP | Header Type | 80h | RO |
| 2Ch–2Fh | SS | Sub System Identifiers | 00000000h | R/WO |
| 34h | CAPP | Capability List Pointer | E0h | RO |
| 40h–43h | PMBASE | ACPI Base Address | 00000001h | R/W, RO |
| 44h | ACPI_CNTL | ACPI Control | 00h | R/W |
| 48h–4Bh | GPIOBASE | GPIO Base Address | 00000001h | R/W, RO |
| 4C | GC | GPIO Control | 00h | R/W |
| 60h–63h | PIRQ[*n*]_ROUT | PIRQ[A–D] Routing Control | 80h | R/W |
| 64h | SIRQ_CNTL | Serial IRQ Control | 10h | R/W, RO |
| 68h–6Bh | PIRQ[*n*]_ROUT | PIRQ[E–H] Routing Control | 80h | R/W |
| 80h | LPC_I/O_DEC | I/O Decode Ranges | 0000h | R/W |

**Table 13-119.LPC Interface PCI Register Address Map (LPC I/F—D31:F0) (Sheet 2 of 2)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 82h–83h | LPC_EN | LPC Interface Enables | 0000h | R/W |
| 84h–87h | GEN1_DEC | LPC Interface Generic Decode Range 1 | 00000000h | R/W |
| 88h–8Bh | GEN2_DEC | LPC Interface Generic Decode Range 2 | 00000000h | R/W |
| 8Ch–8Eh | GEN3_DEC | LPC Interface Generic Decode Range 3 | 00000000h | R/W |
| 90h–93h | GEN4_DEC | LPC Interface Generic Decode Range 4 | 00000000h | R/W |
| A0h–CFh | — | Power Management (See Section 13.8.1) | — | — |
| D0h–D3h | FWH_SEL1 | Firmware Hub Select 1 | 00112233h | R/W, RO |
| D4h–D5h | FWH_SEL2 | Firmware Hub Select 2 | 4567h | R/W |
| D8h–D9h | FWH_DEC_EN1 | Firmware Hub Decode Enable 1 | FFCFh | R/W, RO |
| DCh | BIOS_CNTL | BIOS Control | 00h | R/WLO, R/W |
| E0h–E1h | FDCAP | Feature Detection Capability ID | 0009h | RO |
| E2h | FDLEN | Feature Detection Capability Length | 0Ch | RO |
| E3h | FDVER | Feature Detection Version | 10h | RO |
| E4h–EBh | FDVCT | Feature Vector | See Description | RO |
| F0h–F3h | RCBA | Root Complex Base Address | 00000000h | R/W |

## 13.1.1 VID—Vendor Identification Register (LPC I/F—D31:F0)

Offset Address: 00h–01h
Default Value: 8086h
Lockable: No
Attribute: RO
Size: 16-bit
Power Well: Core

| Bit | Description |
|---|---|
| 15:0 | **Vendor ID — RO**. This is a 16-bit value assigned to Intel. Intel VID = 8086h |

## 13.1.2 DID—Device Identification Register (LPC I/F—D31:F0)

Offset Address: 02h–03h
Default Value: See bit description
Lockable: No
Attribute: RO
Size: 16-bit
Power Well: Core

| Bit | Description |
|---|---|
| 15:0 | **Device ID — RO**. This is a 16-bit value assigned to the Chipset LPC bridge. |

### 13.1.3 PCICMD—PCI COMMAND Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 04h–05h | Attribute: | R/W, RO |
| Default Value: | 0007h | Size: | 16-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:10 | **Reserved** |
| 9 | Fast Back to Back Enable (FBE) — RO. Hardwired to 0. |
| 8 | **SERR# Enable (SERR_EN)** — R/W. The LPC bridge generates SERR# if this bit is set. |
| 7 | Wait Cycle Control (WCC) — RO. Hardwired to 0. |
| 6 | **Parity Error Response Enable (PERE)** — R/W.<br>0 = No action is taken when detecting a parity error.<br>1 = Enables the Chipset LPC bridge to respond to parity errors detected on backbone interface. |
| 5 | **VGA Palette Snoop (VPS) — RO**. Hardwired to 0. |
| 4 | **Memory Write and Invalidate Enable (MWIE) — RO**. Hardwired to 0. |
| 3 | **Special Cycle Enable (SCE) — RO**. Hardwired to 0. |
| 2 | **Bus Master Enable (BME) — RO**. Bus Masters cannot be disabled. |
| 1 | **Memory Space Enable (MSE) — RO**. Memory space cannot be disabled on LPC. |
| 0 | **I/O Space Enable (IOSE) — RO**. I/O space cannot be disabled on LPC. |

### 13.1.4 PCISTS—PCI Status Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 06–07h | Attribute: | RO, R/WC |
| Default Value: | 0210h | Size: | 16-bit |
| Lockable: | Noh | Power Well: | Core |

***Note:*** For the writable bits, software must write a 1 to clear bits that are set. Writing a 0 to the bit has no effect.

| Bit | Description |
|---|---|
| 15 | **Detected Parity Error (DPE) — R/WC.** Set when the LPC bridge detects a parity error on the internal backbone. Set even if the PCICMD.PERE bit (D31:F0:04, bit 6) is 0.<br>0 = Parity Error Not detected.<br>1 = Parity Error detected. |
| 14 | **Signaled System Error (SSE)— R/WC**. Set when the LPC bridge signals a system error to the internal SERR# logic. |
| 13 | **Master Abort Status (RMA) — R/WC**.<br>0 = Unsupported request status not received.<br>1 = The bridge received a completion with unsupported request status from the backbone. |
| 12 | **Received Target Abort (RTA) — R/WC**.<br>0 = Completion abort not received.<br>1 = Completion with completion abort received from the backbone. |

| Bit | Description |
|---|---|
| 11 | **Signaled Target Abort (STA) — R/WC.**<br>0 = Target abort Not generated on the backbone.<br>1 = LPC bridge generated a completion packet with target abort status on the backbone. |
| 10:9 | **DEVSEL# Timing Status (DEV_STS) — RO.**<br>01 = Medium Timing. |
| 8 | **Data Parity Error Detected (DPED) — R/WC.**<br>0 = All conditions listed below Not met.<br>1 = Set when all three of the following conditions are met:<br>• LPC bridge receives a completion packet from the backbone from a previous request,<br>• Parity error has been detected (D31:F0:06, bit 15)<br>• PCICMD.PERE bit (D31:F0:04, bit 6) is set. |
| 7 | **Fast Back to Back Capable (FBC)**: **Reserved** – bit has no meaning on the internal backbone. |
| 6 | **Reserved**. |
| 5 | **66 MHz Capable (66MHZ_CAP)** — **Reserved** – bit has no meaning on internal backbone. |
| 4 | **Capabilities List (CLIST) — RO**. Capability list exists on the LPC bridge. |
| 3 | **Interrupt Status (IS) — RO**. The LPC bridge does not generate interrupts. |
| 2:0 | **Reserved**. |

### 13.1.5 RID—Revision Identification Register (LPC I/F—D31:F0)

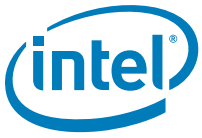
Offset Address: 08h Attribute: RO
Default Value: See bit description Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Revision ID (RID) — RO.** |

### 13.1.6 PI—Programming Interface Register (LPC I/F—D31:F0)

Offset Address: 09h Attribute: RO
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Programming Interface — RO.** |



### 13.1.7 SCC—Sub Class Code Register (LPC I/F—D31:F0)

Offset Address: 0Ah Attribute: RO
Default Value: 01h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Sub Class Code — RO.** 8-bit value that indicates the category of bridge for the LPC bridge.<br>01h = PCI-to-ISA bridge. |

### 13.1.8 BCC—Base Class Code Register (LPC I/F—D31:F0)

Offset Address: 0Bh Attribute: RO
Default Value: 06h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Base Class Code — RO**. 8-bit value that indicates the type of device for the LPC bridge.<br>06h = Bridge device. |

### 13.1.9 PLT—Primary Latency Timer Register (LPC I/F—D31:F0)

Offset Address: 0Dh Attribute: RO
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:3 | **Master Latency Count (MLC)** — **Reserved**. |
| 2:0 | **Reserved**. |

### 13.1.10 HEADTYP—Header Type Register (LPC I/F—D31:F0)

Offset Address: 0Eh Attribute: RO
Default Value: 80h Size: 8 bits

| Bit | Description |
|---|---|
| 7 | **Multi-Function Device — RO**. This bit is 1 to indicate a multi-function device. |
| 6:0 | **Header Type — RO.** This 7-bit field identifies the header layout of the configuration space. |

### 13.1.11 SS—Sub System Identifiers Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 2Ch–2Fh | Attribute: | R/WO |
| Default Value: | 00000000h | Size: | 32 bits |

This register is initialized to logic 0 by the assertion of PLTRST#. This register can be written only once after PLTRST# de-assertion.

| Bit | Description |
|---|---|
| 31:16 | **Subsystem ID (SSID)** — R/WO This is written by BIOS. No hardware action taken on this value. |
| 15:0 | **Subsystem Vendor ID (SSVID)** — R/WO This is written by BIOS. No hardware action taken on this value. |

### 13.1.12 CAPP—Capability List Pointer (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 34h | Attribute: | RO |
| Default Value: | E0h | Size: | 8 bits |
| | | Power Well: | Core |

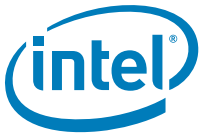
| Bit | Description |
|---|---|
| 7:0 | **Capability Pointer (CP)** — RO. Indicates the offset of the first item. |

### 13.1.13 PMBASE—ACPI Base Address Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 40h–43h | Attribute: | R/W, RO |
| Default Value: | 00000001h | Size: | 32 bit |
| Lockable: | No | Usage: | ACPI, Legacy |
| | | Power Well: | Core |

Sets base address for ACPI I/O registers, GPIO registers and TCO I/O registers. These registers can be mapped anywhere in the 64-K I/O space on 128-byte boundaries.

| Bit | Description |
|---|---|
| 31:16 | **Reserved** |
| 15:7 | **Base Address** — R/W. This field provides 128 bytes of I/O space for ACPI, GPIO, and TCO logic. This is placed on a 128-byte boundary. |
| 6:1 | **Reserved** |
| 0 | Resource Type Indicator (RTE) — RO. Hardwired to 1 to indicate I/O space. |



### 13.1.14 ACPI_CNTL—ACPI Control Register (LPC I/F — D31:F0)

Offset Address: 44h
Default Value: 00h
Lockable: No
Attribute: R/W
Size: 8 bit
Usage: ACPI, Legacy
Power Well: Core

| Bit | Description |
|---|---|
| 7 | **ACPI Enable (ACPI_EN)** — R/W.<br>0 = Disable.<br>1 = Decode of the I/O range pointed to by the ACPI base register is enabled, and the ACPI power management function is enabled. Note that the APM power management ranges (B2/B3h) are always enabled and are not affected by this bit. |
| 6:3 | **Reserved** |
| 2:0 | **SCI IRQ Select (SCI_IRQ_SEL)** — R/W.<br>Specifies on which IRQ the SCI will internally appear. If not using the APIC, the SCI must be routed to IRQ9–11, and that interrupt is not sharable with the SERIRQ stream, but is shareable with other PCI interrupts. If using the APIC, the SCI can also be mapped to IRQ20–23, and can be shared with other interrupts. |

| Bits | SCI Map |
|---|---|
| 000b | IRQ9 |
| 001b | IRQ10 |
| 010b | IRQ11 |
| 011b | **Reserved** |
| 100b | IRQ20 (Only available if APIC enabled) |
| 101b | IRQ21 (Only available if APIC enabled) |
| 110b | IRQ22 (Only available if APIC enabled) |
| 111b | IRQ23 (Only available if APIC enabled) |

**NOTE:** When the interrupt is mapped to APIC interrupts 9, 10 or 11, the APIC should be programmed for active-high reception. When the interrupt is mapped to APIC interrupts 20 through 23, the APIC should be programmed for active-low

### 13.1.15 GPIOBASE—GPIO Base Address Register (LPC I/F — D31:F0)

Offset Address: 48h–4Bh
Default Value: 00000001h
Attribute: R/W, RO
Size: 32 bit

| Bit | Description |
|---|---|
| 31:16 | **Reserved**. Always 0. |
| 15:6 | **Base Address (BA)** — R/W. Provides the 64 bytes of I/O space for GPIO. |
| 5:1 | **Reserved**. Always 0. |
| 0 | RO. Hardwired to 1 to indicate I/O space. |

## 13.1.16 GC—GPIO Control Register (LPC I/F — D31:F0)

Offset Address: 4Ch Attribute: R/W
Default Value: 00h Size: 8 bit

| Bit | Description |
|---|---|
| 7:5 | **Reserved**. |
| 4 | **GPIO Enable (EN)** — R/W. This bit enables/disables decode of the I/O range pointed to by the GPIO Base Address register (D31:F0:48h) and enables the GPIO function.<br>0 = Disable.<br>1 = Enable. |
| 3:0 | **Reserved**. |

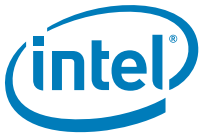
## 13.1.17 PIRQ[n]_ROUT—PIRQ[A,B,C,D] Routing Control Register (LPC I/F—D31:F0)

Offset Address: PIRQA – 60h, PIRQB – 61h, PIRQC – 62h, PIRQD – 63h Attribute:R/W
Default Value: 80h Size:8 bit
Lockable: No Power Well:Core

| Bit | Description |
|---|---|
| 7 | **Interrupt Routing Enable (IRQEN)** — R/W.<br>0 = The corresponding PIRQ is routed to one of the ISA-compatible interrupts specified in bits[3:0].<br>1 = The PIRQ is not routed to the 8259.<br><br>**NOTE:** BIOS must program this bit to 0 during POST for any of the PIRQs that are being used. The value of this bit may subsequently be changed by the OS when setting up for I/O APIC interrupt delivery mode. |
| 6:4 | **Reserved** |
| 3:0 | **IRQ Routing** — R/W. (ISA compatible.) |

| Value | IRQ |
|---|---|
| 0000b | **Reserved** |
| 0001b | **Reserved** |
| 0010b | **Reserved** |
| 0011b | IRQ3 |
| 0100b | IRQ4 |
| 0101b | IRQ5 |
| 0110b | IRQ6 |
| 0111b | IRQ7 |

| Value | IRQ |
|---|---|
| 1000b | **Reserved** |
| 1001b | IRQ9 |
| 1010b | IRQ10 |
| 1011b | IRQ11 |
| 1100b | IRQ12 |
| 1101b | **Reserved** |
| 1110b | IRQ14 |
| 1111b | IRQ15 |



## 13.1.18 SIRQ_CNTL—Serial IRQ Control Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 64h | Attribute: | R/W, RO |
| Default Value: | 10h | Size: | 8 bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7 | **Serial IRQ Enable (SIRQEN)** — R/W.<br>0 = The buffer is input only and internally SERIRQ will be a 1.<br>1 = Serial IRQs will be recognized. The SERIRQ pin will be configured as SERIRQ. |
| 6 | **Serial IRQ Mode Select (SIRQMD)** — R/W.<br>0 = The serial IRQ machine will be in quiet mode.<br>1 = The serial IRQ machine will be in continuous mode.<br><br>**NOTE:** For systems using Quiet Mode, this bit should be set to 1 (Continuous Mode) for at least one frame after coming out of reset before switching back to Quiet Mode. Failure to do so will result in the Chipset not recognizing SERIRQ interrupts. |
| 5:2 | **Serial IRQ Frame Size (SIRQSZ)** — RO. Fixed field that indicates the size of the SERIRQ frame as 21 frames. |
| 1:0 | **Start Frame Pulse Width (SFPW)** — R/W. This is the number of PCI clocks that the SERIRQ pin will be driven low by the serial IRQ machine to signal a start frame. In continuous mode, the Chipset will drive the start frame for the number of clocks specified. In quiet mode, the Chipset will drive the start frame for the number of clocks specified minus one, as the first clock was driven by the peripheral.<br>00 = 4 clocks<br>01 = 6 clocks<br>10 = 8 clocks<br>11 = **Reserved** |

## 13.1.19 PIRQ[n]_ROUT—PIRQ[E,F,G,H] Routing Control Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | PIRQE – 68h, PIRQF – 69h, PIRQG – 6Ah, PIRQH – 6Bh | Attribute: | R/W |
| Default Value: | 80h | Size: | 8 bit |
| Lockable: | No | Power Well: | Core |

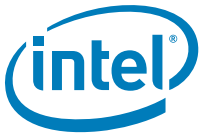
| Bit | Description |
|---|---|
| 7 | **Interrupt Routing Enable (IRQEN)** — R/W.<br>0 = The corresponding PIRQ is routed to one of the ISA-compatible interrupts specified in bits[3:0].<br>1 = The PIRQ is not routed to the 8259.<br><br>**NOTE:** BIOS must program this bit to 0 during POST for any of the PIRQs that are being used. The value of this bit may subsequently be changed by the OS when setting up for I/O APIC interrupt delivery mode. |
| 6:4 | **Reserved** |

| Bit | Description |
|---|---|
| 3:0 | **IRQ Routing** — R/W. (ISA compatible.)<br><br>**Value** / **IRQ**<br>0000b **Reserved**<br>0001b **Reserved**<br>0010b **Reserved**<br>0011b IRQ3<br>0100b IRQ4<br>0101b IRQ5<br>0110b IRQ6<br>0111b IRQ7<br><br>**Value** / **IRQ**<br>1000b **Reserved**<br>1001b IRQ9<br>1010b IRQ10<br>1011b IRQ11<br>1100b IRQ12<br>1101b **Reserved**<br>1110b IRQ14<br>1111b IRQ15 |

## 13.1.20 LPC_I/O_DEC—I/O Decode Ranges Register (LPC I/F—D31:F0)

Offset Address: 80h Attribute: R/W
Default Value: 0000h Size: 16 bit

| Bit | Description |
|---|---|
| 15:13 | **Reserved** |
| 12 | **FDD Decode Range** — R/W. Determines which range to decode for the FDD Port<br>0 = 3F0h – 3F5h, 3F7h (Primary)<br>1 = 370h – 375h, 377h (Secondary) |
| 11:10 | **Reserved** |
| 9:8 | **LPT Decode Range** — R/W. This field determines which range to decode for the LPT Port.<br>00 = 378h – 37Fh and 778h – 77Fh<br>01 = 278h – 27Fh (port 279h is read only) and 678h – 67Fh<br>10 = 3BCh –3BEh and 7BCh – 7BEh<br>11 = **Reserved** |
| 7 | **Reserved** |
| 6:4 | **COMB Decode Range** — R/W. This field determines which range to decode for the COMB Port.<br>000 = 3F8h – 3FFh (COM1)<br>001 = 2F8h – 2FFh (COM2)<br>010 = 220h – 227h<br>011 = 228h – 22Fh<br>100 = 238h – 23Fh<br>101 = 2E8h – 2EFh (COM4)<br>110 = 338h – 33Fh<br>111 = 3E8h – 3EFh (COM3) |
| 3 | **Reserved** |



| Bit | Description |
|---|---|
| 2:0 | **COMA Decode Range** — R/W. This field determines which range to decode for the COMA Port.<br>000 = 3F8h – 3FFh (COM1)<br>001 = 2F8h – 2FFh (COM2)<br>010 = 220h – 227h<br>011 = 228h – 22Fh<br>100 = 238h – 23Fh<br>101 = 2E8h – 2EFh (COM4)<br>110 = 338h – 33Fh<br>111 = 3E8h – 3EFh (COM3) |

## 13.1.21 LPC_EN—LPC I/F Enables Register (LPC I/F—D31:F0)

Offset Address: 82h – 83h
Default Value: 0000h
Attribute: R/W
Size: 16 bit
Power Well: Core

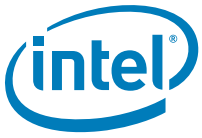
| Bit | Description |
|---|---|
| 15:14 | **Reserved** |
| 13 | **CNF2_LPC_EN** — R/W. Microcontroller Enable # 2.<br>0 = Disable.<br>1 = Enables the decoding of the I/O locations 4Eh and 4Fh to the LPC interface. This range is used for a microcontroller. |
| 12 | **CNF1_LPC_EN** — R/W. Super I/O Enable.<br>0 = Disable.<br>1 = Enables the decoding of the I/O locations 2Eh and 2Fh to the LPC interface. This range is used for Super I/O devices. |
| 11 | **MC_LPC_EN** — R/W. Microcontroller Enable # 1.<br>0 = Disable.<br>1 = Enables the decoding of the I/O locations 62h and 66h to the LPC interface. This range is used for a microcontroller. |
| 10 | **KBC_LPC_EN** — R/W. Keyboard Enable.<br>0 = Disable.<br>1 = Enables the decoding of the I/O locations 60h and 64h to the LPC interface. This range is used for a microcontroller. |
| 9 | **GAMEH_LPC_EN** — R/W. High Gameport Enable<br>0 = Disable.<br>1 = Enables the decoding of the I/O locations 208h to 20Fh to the LPC interface. This range is used for a gameport. |
| 8 | **GAMEL_LPC_EN** — R/W. Low Gameport Enable<br>0 = Disable.<br>1 = Enables the decoding of the I/O locations 200h to 207h to the LPC interface. This range is used for a gameport. |
| 7:4 | **Reserved** |

| Bit | Description |
|---|---|
| 3 | **FDD_LPC_EN** — R/W. Floppy Drive Enable<br>0 = Disable.<br>1 = Enables the decoding of the FDD range to the LPC interface. This range is selected in the LPC_FDD/LPT Decode Range Register (D31:F0:80h, bit 12). |
| 2 | **LPT_LPC_EN** — R/W. Parallel Port Enable<br>0 = Disable.<br>1 = Enables the decoding of the LPTrange to the LPC interface. This range is selected in the LPC_FDD/LPT Decode Range Register (D31:F0:80h, bit 9:8). |
| 1 | **COMB_LPC_EN** — R/W. Com Port B Enable<br>0 = Disable.<br>1 = Enables the decoding of the COMB range to the LPC interface. This range is selected in the LPC_COM Decode Range Register (D31:F0:80h, bits 6:4). |
| 0 | **COMA_LPC_EN** — R/W. Com Port A Enable<br>0 = Disable.<br>1 = Enables the decoding of the COMA range to the LPC interface. This range is selected in the LPC_COM Decode Range Register (D31:F0:80h, bits 3:2). |

### 13.1.22 GEN1_DEC—LPC I/F Generic Decode Range 1 Register (LPC I/F—D31:F0)

Offset Address: 84h – 87h
Default Value: 00000000h
Attribute: R/W
Size: 32 bit
Power Well: Core

| Bit | Description |
|---|---|
| 31:24 | **Reserved** |
| 23:18 | **Generic I/O Decode Range Address[7:2] Mask**: A ‘1’ in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size. |
| 17:16 | **Reserved** |
| 15:2 | **Generic I/O Decode Range 1 Base Address (GEN1_BASE) — R/W**. This address is aligned on a 128-byte boundary, and must have address lines 31:16 as 0.<br><br>**NOTE:** The Chipset does not provide decode down to the word or byte level. |
| 1 | **Reserved** |
| 0 | **Generic Decode Range 1 Enable (GEN1_EN) — R/W.**<br>0 = Disable.<br>1 = Enable the GEN1 I/O range to be forwarded to the LPC I/F |



### 13.1.23 GEN2_DEC—LPC I/F Generic Decode Range 2Register (LPC I/F—D31:F0)

Offset Address: 88h – 8Bh
Default Value: 00000000h
Attribute: R/W
Size: 32 bit
Power Well: Core

| Bit | Description |
|---|---|
| 31:24 | **Reserved** |
| 23:18 | **Generic I/O Decode Range Address[7:2] Mask**: A '1' in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size. |
| 17:16 | **Reserved** |
| 15:2 | **Generic I/O Decode Range 2Base Address (GEN1_BASE) — R/W**.<br><br>**NOTE:** The Chipset does not provide decode down to the word or byte level. |
| 1 | **Reserved** |
| 0 | **Generic Decode Range 2Enable (GEN2_EN) — R/W**.<br>0 = Disable.<br>1 = Enable the GEN2 I/O range to be forwarded to the LPC I/F |

### 13.1.24 GEN3_DEC—LPC I/F Generic Decode Range 3Register (LPC I/F—D31:F0)

Offset Address: 8Ch – 8Eh
Default Value: 00000000h
Attribute: R/W
Size: 32 bit
Power Well: Core

| Bit | Description |
|---|---|
| 31:24 | **Reserved** |
| 23:18 | **Generic I/O Decode Range Address[7:2] Mask:** A '1' in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size. |
| 17:16 | **Reserved** |
| 15:2 | **Generic I/O Decode Range 3Base Address (GEN3_BASE) — R/W.**<br>**NOTE:** The Chipset does not provide decode down to the word or byte level. |
| 1 | **Reserved** |
| 0 | **Generic Decode Range 3Enable (GEN3_EN) — R/W.**<br>0 = Disable.<br>1 = Enable the GEN3 I/O range to be forwarded to the LPC I/F |

### 13.1.25 GEN4_DEC—LPC I/F Generic Decode Range 4Register (LPC I/F—D31:F0)

Offset Address: 90h – 93h
Default Value: 00000000h
Attribute: R/W
Size: 32 bit
Power Well: Core

| Bit | Description |
|---|---|
| 31:24 | **Reserved** |
| 23:18 | **Generic I/O Decode Range Address[7:2] Mask**: A '1' in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size. |
| 17:16 | **Reserved** |
| 15:2 | **Generic I/O Decode Range 4Base Address (GEN4_BASE) — R/W.**<br>**NOTE:** The Chipset does not provide decode down to the word or byte level. |
| 1 | **Reserved** |
| 0 | **Generic Decode Range 4Enable (GEN4_EN) — R/W.**<br>0 = Disable.<br>1 = Enable the GEN4 I/O range to be forwarded to the LPC I/F |

### 13.1.26 FWH_SEL1—Firmware Hub Select 1 Register (LPC I/F—D31:F0)

Offset Address: D0h–D3h
Default Value: 00112233h
Attribute: R/W, RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:28 | **FWH_F8_IDSEL** — RO. IDSEL for two 512-KB Firmware Hub memory ranges and one 128-KB memory range. This field is fixed at 0000. The IDSEL programmed in this field addresses the following memory ranges:<br>FFF8 0000h – FFFF FFFFh<br>FFB8 0000h – FFBF FFFFh<br>000E 0000h – 000F FFFFh |
| 27:24 | **FWH_F0_IDSEL** — R/W. IDSEL for two 512-KB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFF0 0000h – FFF7 FFFFh<br>FFB0 0000h – FFB7 FFFFh |
| 23:20 | **FWH_E8_IDSEL** — R/W. IDSEL for two 512-KB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFE8 0000h – FFEF FFFFh<br>FFA8 0000h – FFAF FFFFh |
| 19:16 | **FWH_E0_IDSEL** — R/W. IDSEL for two 512-KB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFE0 0000h – FFE7 FFFFh<br>FFA0 0000h – FFA7 FFFFh |

| Bit | Description |
|---|---|
| 15:12 | **FWH_D8_IDSEL** — R/W. IDSEL for two 512-KB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFD8 0000h – FFDF FFFFh<br>FF98 0000h – FF9F FFFFh |
| 11:8 | **FWH_D0_IDSEL** — R/W. IDSEL for two 512-KB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFD0 0000h – FFD7 FFFFh<br>FF90 0000h – FF97 FFFFh |
| 7:4 | **FWH_C8_IDSEL** — R/W. IDSEL for two 512-KB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFC8 0000h – FFCF FFFFh<br>FF88 0000h – FF8F FFFFh |
| 3:0 | **FWH_C0_IDSEL** — R/W. IDSEL for two 512-KB Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFC0 0000h – FFC7 FFFFh<br>FF80 0000h – FF87 FFFFh |

## 13.1.27 FWH_SEL2—Firmware Hub Select 2 Register (LPC I/F—D31:F0)

Offset Address: D4h–D5h Attribute: R/W
Default Value: 4567h Size: 16 bits

| Bit | Description |
|---|---|
| 15:12 | **FWH_70_IDSEL** — R/W. IDSEL for two, 1-M Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FF70 0000h – FF7F FFFFh<br>FF30 0000h – FF3F FFFFh |
| 11:8 | **FWH_60_IDSEL** — R/W. IDSEL for two, 1-M Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FF60 0000h – FF6F FFFFh<br>FF20 0000h – FF2F FFFFh |
| 7:4 | **FWH_50_IDSEL** — R/W. IDSEL for two, 1-M Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FF50 0000h – FF5F FFFFh<br>FF10 0000h – FF1F FFFFh |
| 3:0 | **FWH_40_IDSEL** — R/W. IDSEL for two, 1-M Firmware Hub memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FF40 0000h – FF4F FFFFh<br>FF00 0000h – FF0F FFFFh |

## 13.1.28 FWH_DEC_EN1—Firmware Hub Decode Enable Register (LPC I/F—D31:F0)

Offset Address: D8h–D9h Attribute: R/W, RO
Default Value: FFCFh Size: 16 bits

| Bit | Description |
|---|---|
| 15 | **FWH_F8_EN** — RO. This bit enables decoding two 512-KB Firmware Hub memory ranges, and one 128-KB memory range.<br>0 = Disable<br>1 = Enable the following ranges for the Firmware Hub<br>FFF80000h – FFFFFFFFh<br>FFB80000h – FFBFFFFFh |
| 14 | **FWH_F0_EN** — R/W. This bit enables decoding two 512-KB Firmware Hub memory ranges.<br>0 = Disable.<br>1 = Enable the following ranges for the Firmware Hub:<br>FFF00000h – FFF7FFFFh<br>FFB00000h – FFB7FFFFh |
| 13 | **FWH_E8_EN** — R/W. This bit enables decoding two 512-KB Firmware Hub memory ranges.<br>0 = Disable.<br>1 = Enable the following ranges for the Firmware Hub:<br>FFE80000h – FFEFFFFFh<br>FFA80000h – FFAFFFFFh |
| 12 | **FWH_E0_EN** — R/W. This bit enables decoding two 512-KB Firmware Hub memory ranges.<br>0 = Disable.<br>1 = Enable the following ranges for the Firmware Hub:<br>FFE00000h – FFE7FFFFh<br>FFA00000h – FFA7FFFFh |
| 11 | **FWH_D8_EN** — R/W. This bit enables decoding two 512-KB Firmware Hub memory ranges.<br>0 = Disable.<br>1 = Enable the following ranges for the Firmware Hub<br>FFD80000h – FFDFFFFFh<br>FF980000h – FF9FFFFFh |
| 10 | **FWH_D0_EN** — R/W. This bit enables decoding two 512-KB Firmware Hub memory ranges.<br>0 = Disable.<br>1 = Enable the following ranges for the Firmware Hub<br>FFD00000h – FFD7FFFFh<br>FF900000h – FF97FFFFh |
| 9 | **FWH_C8_EN** — R/W. This bit enables decoding two 512-KB Firmware Hub memory ranges.<br>0 = Disable.<br>1 = Enable the following ranges for the Firmware Hub<br>FFC80000h – FFCFFFFFh<br>FF880000h – FF8FFFFFh |

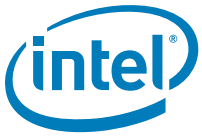
| Bit | Description |
|---|---|
| 8 | **FWH_C0_EN** — R/W. This bit enables decoding two 512-KB Firmware Hub memory ranges.<br>0 = Disable.<br>1 = Enable the following ranges for the Firmware Hub<br>FFC00000h – FFC7FFFFh<br>FF800000h – FF87FFFFh |
| 7 | **FWH_Legacy_F_EN** — R/W. This enables the decoding of the legacy 128-K range at F0000h – FFFFFh.<br>0 = Disable.<br>1 = Enable the following legacy ranges for the Firmware Hub<br>F0000h – FFFFFh |
| 6 | **FWH_Legacy_E_EN** — R/W. This enables the decoding of the legacy 128-K range at E0000h – EFFFFh.<br>0 = Disable.<br>1 = Enable the following legacy ranges for the Firmware Hub<br>E0000h – EFFFFh |
| 5:4 | **Reserved** |
| 3 | **FWH_70_EN** — R/W. Enables decoding two 1-M Firmware Hub memory ranges.<br>0 = Disable.<br>1 = Enable the following ranges for the Firmware Hub<br>FF70 0000h – FF7F FFFFh<br>FF30 0000h – FF3F FFFFh |
| 2 | **FWH_60_EN** — R/W. Enables decoding two 1-M Firmware Hub memory ranges.<br>0 = Disable.<br>1 = Enable the following ranges for the Firmware Hub<br>FF60 0000h – FF6F FFFFh<br>FF20 0000h – FF2F FFFFh |
| 1 | **FWH_50_EN** — R/W. Enables decoding two 1-M Firmware Hub memory ranges.<br>0 = Disable.<br>1 = Enable the following ranges for the Firmware Hub<br>FF50 0000h – FF5F FFFFh<br>FF10 0000h – FF1F FFFFh |
| 0 | **FWH_40_EN** — R/W. Enables decoding two 1-M Firmware Hub memory ranges.<br>0 = Disable.<br>1 = Enable the following ranges for the Firmware Hub<br>FF40 0000h – FF4F FFFFh<br>FF00 0000h – FF0F FFFFh |

***Note:*** This register effects the BIOS decode regardless of whether the BIOS is resident on LPC or SPI. The concept of Feature Space does not apply to SPI-based flash. The chipset simply decodes these ranges as memory accesses when enabled for the SPI flash interface.

## 13.1.29 BIOS_CNTL—BIOS Control Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | DCh | Attribute: | R/WLO, R/W, RO |
| Default Value: | 00h | Size: | 8 bit |
| Lockable: | No | Power Well: | Core |

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7:5</td><td>Reserved</td></tr>
<tr><td>4</td><td>Top Swap Status (TSS)— RO: This bit provides a read-only path to view the state of the Top Swap bit that is at offset 3414h, bit 0.</td></tr>
<tr><td>3:2</td><td>SPI Read Configuration (SRC)— R/W: This 2-bit field controls two policies related to BIOS reads on the SPI interface:<br>Bit 3- Prefetch Enable<br>Bit 2- Cache Disable<br>Settings are summarized below:
<table>
<tr><th>Bits 3:2</th><th>Description</th></tr>
<tr><td>00b</td><td>No prefetching, but caching enabled. 64B demand reads load the read buffer cache with “valid” data, allowing repeated code fetches to the same line to complete quickly</td></tr>
<tr><td>01b</td><td>No prefetching and no caching. One-to-one correspondence of host BIOS reads to SPI cycles. This value can be used to invalidate the cache.</td></tr>
<tr><td>10b</td><td>Prefetching and Caching enabled. This mode is used for long sequences of short reads to consecutive addresses (i.e., shadowing).</td></tr>
<tr><td>11b</td><td>Reserved. This is an invalid configuration, caching must be enabled when prefetching is enabled.</td></tr>
</table></td></tr>
<tr><td>1</td><td>BIOS Lock Enable (BLE) — R/WLO.<br>0 = Setting the BIOSWE will not cause SMIs.<br>1 = Enables setting the BIOSWE bit to cause SMIs. Once set, this bit can only be cleared by a PLTRST#</td></tr>
<tr><td>0</td><td>BIOS Write Enable (BIOSWE) — R/W.<br>0 = Only read cycles permitted to Firmware Hub or SPI flash.<br>1 = Access to the BIOS space is enabled for both read and write cycles. When this bit is written from a 0 to a 1 and BIOS Lock Enable (BLE) is also set, an SMI# is generated. This ensures that only SMI code can update BIOS.<br><br>NOTE: Writes to the Firmware Hub’s Feature Space are not blocked when the BIOSWE is cleared in order to allow access to registers. The Feature Space is the second range that is located 4 MB below the BIOS range for each Firmware Hub.</td></tr>
</table>



### 13.1.30 FDCAP—Feature Detection Capability ID (LPC I/F—D31:F0)

Offset Address: E0h-E1h
Default Value: 0009h
Attribute: RO
Size: 16 bit
Power Well: Core

| Bit | Description |
|---|---|
| 15:8 | **Next Item Pointer (NEXT)**: Configuration offset of the next Capability Item. 00h indicates the last item in the Capability List. |
| 7:0 | Capability ID: Indicates a Vendor Specific Capability |

### 13.1.31 FDLEN—Feature Detection Capability Length (LPC I/F—D31:F0)

Offset Address: E2h
Default Value: 0Ch
Attribute: RO
Size: 8 bit
Power Well: Core

| Bit | Description |
|---|---|
| 7:0 | Capability Length: Indicates the length of this Vendor Specific capability, as required by PCI Spec. |

### 13.1.32 FDVER—Feature Detection Version (LPC I/F—D31:F0)

Offset Address: E3h
Default Value: 10h
Attribute: RO
Size: 8 bit
Power Well: Core

| Bit | Description |
|---|---|
| 7:4 | **Vendor-Specific Capability ID**: A value of 1h in this 4-bit field identifies this Capability as Feature Detection Type. This field allows software to differentiate the Feature Detection Capability from other Vendor-Specific capabilities |
| 3:0 | **Capability Version**: This field indicates the version of the Feature Detection capability |

### 13.1.33 FDVCT—Feature Vector Register (LPC I/F—D31:F0)

Offset Address: E4h–EBh
Default Value: See Description
Attribute: RO
Size: 64 bit
Power Well: Core

| Bit | Description |
|---|---|
| 63:10 | **Reserved** |
| 9 | **Mobile Features Capability— RO**:<br>0 = Disabled<br>1 = Capable |

| Bit | Description |
|---|---|
| 8:4 | **Reserved** |
| 3 | **SATA AHCI Capability— RO**:<br>0 = Capable<br>1 = Disabled |
| 2:0 | **Reserved** |

### 13.1.34 RCBA—Root Complex Base Address Register (LPC I/F—D31:F0)

Offset Address: F0h Attribute: R/W
Default Value: 00000000h Size: 32 bit

| Bit | Description |
|---|---|
| 31:14 | **Base Address (BA)** — R/W. Base Address for the root complex register block decode range. This address is aligned on a 16-KB boundary. |
| 13:1 | **Reserved** |
| 0 | **Enable (EN)** — R/W. When set, this bit enables the range specified in BA to be claimed as the Root Complex Register Block. |

## 13.2 DMA I/O Registers (LPC I/F—D31:F0)

**Table 13-120.DMA Registers (Sheet 1 of 2)**

| Port | Alias | Register Name | Default | Type |
|---|---|---|---|---|
| 00h | 10h | Channel 0 DMA Base & Current Address | Undefined | R/W |
| 01h | 11h | Channel 0 DMA Base & Current Count | Undefined | R/W |
| 02h | 12h | Channel 1 DMA Base & Current Address | Undefined | R/W |
| 03h | 13h | Channel 1 DMA Base & Current Count | Undefined | R/W |
| 04h | 14h | Channel 2 DMA Base & Current Address | Undefined | R/W |
| 05h | 15h | Channel 2 DMA Base & Current Count | Undefined | R/W |
| 06h | 16h | Channel 3 DMA Base & Current Address | Undefined | R/W |
| 07h | 17h | Channel 3 DMA Base & Current Count | Undefined | R/W |
| 08h | 18h | Channel 0–3 DMA Command | Undefined | WO |
| | | Channel 0–3 DMA Status | Undefined | RO |
| 0Ah | 1Ah | Channel 0–3 DMA Write Single Mask | 000001XXb | WO |
| 0Bh | 1Bh | Channel 0–3 DMA Channel Mode | 000000XXb | WO |
| 0Ch | 1Ch | Channel 0–3 DMA Clear Byte Pointer | Undefined | WO |
| 0Dh | 1Dh | Channel 0–3 DMA Master Clear | Undefined | WO |
| 0Eh | 1Eh | Channel 0–3 DMA Clear Mask | Undefined | WO |
| 0Fh | 1Fh | Channel 0–3 DMA Write All Mask | 0Fh | R/W |
| 80h | 90h | **Reserved** Page | Undefined | R/W |

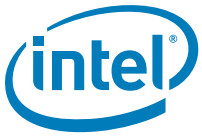


**Table 13-120.DMA Registers (Sheet 2 of 2)**

| Port | Alias | Register Name | Default | Type |
|---|---|---|---|---|
| 81h | 91h | Channel 2 DMA Memory Low Page | Undefined | R/W |
| 82h | — | Channel 3 DMA Memory Low Page | Undefined | R/W |
| 83h | 93h | Channel 1 DMA Memory Low Page | Undefined | R/W |
| 84h–86h | 94h–96h | **Reserved** Pages | Undefined | R/W |
| 87h | 97h | Channel 0 DMA Memory Low Page | Undefined | R/W |
| 88h | 98h | **Reserved** Page | Undefined | R/W |
| 89h | 99h | Channel 6 DMA Memory Low Page | Undefined | R/W |
| 8Ah | 9Ah | Channel 7 DMA Memory Low Page | Undefined | R/W |
| 8Bh | 9Bh | Channel 5 DMA Memory Low Page | Undefined | R/W |
| 8Ch–8Eh | 9Ch–9Eh | **Reserved** Page | Undefined | R/W |
| 8Fh | 9Fh | Refresh Low Page | Undefined | R/W |
| C0h | C1h | Channel 4 DMA Base & Current Address | Undefined | R/W |
| C2h | C3h | Channel 4 DMA Base & Current Count | Undefined | R/W |
| C4h | C5h | Channel 5 DMA Base & Current Address | Undefined | R/W |
| C6h | C7h | Channel 5 DMA Base & Current Count | Undefined | R/W |
| C8h | C9h | Channel 6 DMA Base & Current Address | Undefined | R/W |
| CAh | CBh | Channel 6 DMA Base & Current Count | Undefined | R/W |
| CCh | CDh | Channel 7 DMA Base & Current Address | Undefined | R/W |
| CEh | CFh | Channel 7 DMA Base & Current Count | Undefined | R/W |
| D0h | D1h | Channel 4–7 DMA Command | Undefined | WO |
| | | Channel 4–7 DMA Status | Undefined | RO |
| D4h | D5h | Channel 4–7 DMA Write Single Mask | 000001XXb | WO |
| D6h | D7h | Channel 4–7 DMA Channel Mode | 000000XXb | WO |
| D8h | D9h | Channel 4–7 DMA Clear Byte Pointer | Undefined | WO |
| DAh | DBh | Channel 4–7 DMA Master Clear | Undefined | WO |
| DCh | DDh | Channel 4–7 DMA Clear Mask | Undefined | WO |
| DEh | DFh | Channel 4–7 DMA Write All Mask | 0Fh | R/W |

## 13.2.1 DMABASE_CA—DMA Base and Current Address Registers (LPC I/F—D31:F0)

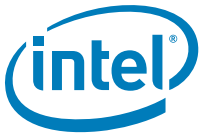
I/O Address: Ch. #0 = 00h; Ch. #1 = 02h Ch. #2 = 04h; Ch. #3 = 06h Ch. #5 = C4h Ch. #6 = C8h Ch. #7 = CCh;
Attribute:R/W
Size:16 bit (per channel), but accessed in two 8-bit quantities
Default Value: Undef
Lockable: No
Power Well:Core

| Bit | Description |
|---|---|
| 15:0 | **Base and Current Address** — R/W. This register determines the address for the transfers to be performed. The address specified points to two separate registers. On writes, the value is stored in the *Base Address* register and copied to the *Current Address* register. On reads, the value is returned from the *Current Address* register.<br>The address increments/decrements in the Current Address register after each transfer, depending on the mode of the transfer. If the channel is in auto-initialize mode, the Current Address register will be reloaded from the Base Address register after a terminal count is generated.<br>For transfers to/from a 16-bit slave (channel's 5-7), the address is shifted left one bit location. Bit 15 will be shifted into Bit 16.<br>The register is accessed in 8 bit quantities. The byte is pointed to by the current byte pointer flip/flop. Before accessing an address register, the byte pointer flip/flop should be cleared to ensure that the low byte is accessed first. |

### 13.2.2 DMABASE_CC—DMA Base and Current Count Registers (LPC I/F—D31:F0)

I/O Address: Ch. #0 = 01h; Ch. #1 = 03h; Ch. #2 = 05h; Ch. #3 = 07h; Ch. #5 = C6h; Ch. #6 = CAh; Ch. #7 = CEh
Attribute:R/W
Size:16-bit (per channel), but accessed in two 8-bit quantities
Default Value: Undefined
Lockable: No
Power Well:Core

| Bit | Description |
|---|---|
| 15:0 | **Base and Current Count** — R/W. This register determines the number of transfers to be performed. The address specified points to two separate registers. On writes, the value is stored in the *Base Count* register and copied to the *Current Count* register. On reads, the value is returned from the *Current Count* register.<br>The actual number of transfers is one more than the number programmed in the Base Count Register (i.e., programming a count of 4h results in 5 transfers). The count is decrements in the Current Count register after each transfer. When the value in the register rolls from 0 to FFFFh, a terminal count is generated. If the channel is in auto-initialize mode, the Current Count register will be reloaded from the Base Count register after a terminal count is generated.<br>For transfers to/from an 8-bit slave (channels 0–3), the count register indicates the number of bytes to be transferred. For transfers to/from a 16-bit slave (channels 5–7), the count register indicates the number of words to be transferred.<br>The register is accessed in 8 bit quantities. The byte is pointed to by the current byte pointer flip/flop. Before accessing a count register, the byte pointer flip/flop should be cleared to ensure that the low byte is accessed first. |



### 13.2.3 DMAMEM_LP—DMA Memory Low Page Registers (LPC I/F—D31:F0)

I/O Address: Ch. #0 = 87h; Ch. #1 = 83h
Ch. #2 = 81h; Ch. #3 = 82h
Ch. #5 = 8Bh; Ch. #6 = 89h
Ch. #7 = 8Ah; Attribute: R/W
Default Value: Undefined Size: 8-bit
Lockable: No Power Well: Core

| Bit | Description |
|---|---|
| 7:0 | **DMA Low Page** (ISA Address bits [23:16]) — R/W. This register works in conjunction with the DMA controller's Current Address Register to define the complete 24-bit address for the DMA channel. This register remains static throughout the DMA transfer. Bit 16 of this register is ignored when in 16 bit I/O count by words mode as it is replaced by the bit 15 shifted out from the current address register. |

### 13.2.4 DMACMD—DMA Command Register (LPC I/F—D31:F0)

I/O Address: Ch. #0–3 = 08h;
Ch. #4–7 = D0h Attribute: WO
Default Value: Undefined Size: 8-bit
Lockable: No Power Well: Core

| Bit | Description |
|---|---|
| 7:5 | **Reserved**. Must be 0. |
| 4 | **DMA Group Arbitration Priority** — WO. Each channel group is individually assigned either fixed or rotating arbitration priority. At part reset, each group is initialized in fixed priority.<br>0 = Fixed priority to the channel group<br>1 = Rotating priority to the group. |
| 3 | **Reserved**. Must be 0. |
| 2 | **DMA Channel Group Enable** — WO. Both channel groups are enabled following part reset.<br>0 = Enable the DMA channel group.<br>1 = Disable. Disabling channel group 4–7 also disables channel group 0–3, which is cascaded through channel 4. |
| 1:0 | **Reserved**. Must be 0. |

### 13.2.5 DMASTA—DMA Status Register (LPC I/F—D31:F0)

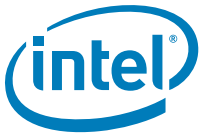
| | | | |
|---|---|---|---|
| I/O Address: | Ch. #0–3 = 08h;<br>Ch. #4–7 = D0h | Attribute: | RO |
| Default Value: | Undefined | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:4 | **Channel Request Status** — RO. When a valid DMA request is pending for a channel, the corresponding bit is set to 1. When a DMA request is not pending for a particular channel, the corresponding bit is set to 0. The source of the DREQ may be hardware or a software request. Note that channel 4 is the cascade channel, so the request status of channel 4 is a logical OR of the request status for channels 0 through 3.<br>4 = Channel 0<br>5 = Channel 1 (5)<br>6 = Channel 2 (6)<br>7 = Channel 3 (7) |
| 3:0 | **Channel Terminal Count Status** — RO. When a channel reaches terminal count (TC), its status bit is set to 1. If TC has not been reached, the status bit is set to 0. Channel 4 is programmed for cascade, so the TC bit response for channel 4 is irrelevant:<br>0 = Channel 0<br>1 = Channel 1 (5)<br>2 = Channel 2 (6)<br>3 = Channel 3 (7) |

### 13.2.6 DMA_WRSMSK—DMA Write Single Mask Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | Ch. #0–3 = 0Ah;<br>Ch. #4–7 = D4h | Attribute: | WO |
| Default Value: | 0000 01xx | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:3 | **Reserved**. Must be 0. |
| 2 | **Channel Mask Select** — WO.<br>0 = Enable DREQ for the selected channel. The channel is selected through bits [1:0]. Therefore, only one channel can be masked / unmasked at a time.<br>1 = Disable DREQ for the selected channel. |
| 1:0 | **DMA Channel Select** — WO. These bits select the DMA Channel Mode Register to program.<br>00 = Channel 0 (4)<br>01 = Channel 1 (5)<br>10 = Channel 2 (6)<br>11 = Channel 3 (7) |



## 13.2.7 DMACH_MODE—DMA Channel Mode Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | Ch. #0–3 = 0Bh;<br>Ch. #4–7 = D6h | Attribute: | WO |
| Default Value: | 0000 00xx | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:6 | **DMA Transfer Mode** — WO. Each DMA channel can be programmed in one of four different modes:<br>00 = Demand mode<br>01 = Single mode<br>10 = **Reserved**<br>11 = Cascade mode |
| 5 | **Address Increment/Decrement Select** — WO. This bit controls address increment/decrement during DMA transfers.<br>0 = Address increment. (default after part reset or Master Clear)<br>1 = Address decrement. |
| 4 | **Autoinitialize Enable** — WO.<br>0 = Autoinitialize feature is disabled and DMA transfers terminate on a terminal count. A part reset or Master Clear disables autoinitialization.<br>1 = DMA restores the Base Address and Count registers to the current registers following a terminal count (TC). |
| 3:2 | **DMA Transfer Type** — WO. These bits represent the direction of the DMA transfer. When the channel is programmed for cascade mode, (bits[7:6] = 11) the transfer type is irrelevant.<br>00 = Verify – No I/O or memory strobes generated<br>01 = Write – Data transferred from the I/O devices to memory<br>10 = Read – Data transferred from memory to the I/O device<br>11 = Invalid |
| 1:0 | **DMA Channel Select** — WO. These bits select the DMA Channel Mode Register that will be written by bits [7:2].<br>00 = Channel 0 (4)<br>01 = Channel 1 (5)<br>10 = Channel 2 (6)<br>11 = Channel 3 (7) |

### 13.2.8 DMA Clear Byte Pointer Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | Ch. #0–3 = 0Ch;<br>Ch. #4–7 = D8h | Attribute: | WO |
| Default Value: | xxxx xxxx | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | **Clear Byte Pointer** — WO. No specific pattern. Command enabled with a write to the I/O port address. Writing to this register initializes the byte pointer flip/flop to a known state. It clears the internal latch used to address the upper or lower byte of the 16-bit Address and Word Count Registers. The latch is also cleared by part reset and by the Master Clear command. This command precedes the first access to a 16-bit DMA controller register. The first access to a 16-bit register will then access the significant byte, and the second access automatically accesses the most significant byte. |

### 13.2.9 DMA Master Clear Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | Ch. #0–3 = 0Dh;<br>Ch. #4–7 = DAh | Attribute: | WO |
| Default Value: | xxxx xxxx | Size: | 8-bit |

| Bit | Description |
|---|---|
| 7:0 | **Master Clear** — WO. No specific pattern. Enabled with a write to the port. This has the same effect as the hardware Reset. The Command, Status, Request, and Byte Pointer flip/flop registers are cleared and the Mask Register is set. |

### 13.2.10 DMA_CLMSK—DMA Clear Mask Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | Ch. #0–3 = 0Eh;<br>Ch. #4–7 = DCh | Attribute: | WO |
| Default Value: | xxxx xxxx | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | **Clear Mask Register** — WO. No specific pattern. Command enabled with a write to the port. |

### 13.2.11 DMA_WRMSK—DMA Write All Mask Register (LPC I/F—D31:F0)

I/O Address: Ch. #0–3 = 0Fh; Ch. #4–7 = DEh
Attribute: R/W
Default Value: 0000 1111
Size: 8-bit
Lockable: No
Power Well: Core

| Bit | Description |
|---|---|
| 7:4 | **Reserved**. Must be 0. |
| 3:0 | **Channel Mask Bits** — R/W. This register permits all four channels to be simultaneously enabled/disabled instead of enabling/disabling each channel individually, as is the case with the Mask Register – Write Single Mask Bit. In addition, this register has a read path to allow the status of the channel mask bits to be read. A channel's mask bit is automatically set to 1 when the Current Byte/Word Count Register reaches terminal count (unless the channel is in auto-initialization mode).<br>Setting the bit(s) to a 1 disables the corresponding DREQ(s). Setting the bit(s) to a 0 enables the corresponding DREQ(s). Bits [3:0] are set to 1 upon part reset or Master Clear. When read, bits [3:0] indicate the DMA channel [3:0] ([7:4]) mask status.<br>Bit 0 = Channel 0 (4)1 = Masked, 0 = Not Masked<br>Bit 1 = Channel 1 (5)1 = Masked, 0 = Not Masked<br>Bit 2 = Channel 2 (6)1 = Masked, 0 = Not Masked<br>Bit 3 = Channel 3 (7)1 = Masked, 0 = Not Masked<br><br>**NOTE:** Disabling channel 4 also disables channels 0–3 due to the cascade of channel's 0 – 3 through channel 4. |

## 13.3 Timer I/O Registers (LPC I/F—D31:F0)

| Port | Aliases | Register Name | Default Value | Type |
|---|---|---|---|---|
| 40h | 50h | Counter 0 Interval Time Status Byte Format | 0XXXXXXXb | RO |
| | | Counter 0 Counter Access Port | Undefined | R/W |
| 41h | 51h | Counter 1 Interval Time Status Byte Format | 0XXXXXXXb | RO |
| | | Counter 1 Counter Access Port | Undefined | R/W |
| 42h | 52h | Counter 2 Interval Time Status Byte Format | 0XXXXXXXb | RO |
| | | Counter 2 Counter Access Port | Undefined | R/W |
| 43h | 53h | Timer Control Word | Undefined | WO |
| | | Timer Control Word Register | XXXXXXX0b | WO |
| | | Counter Latch Command | X0h | WO |

### 13.3.1 TCW—Timer Control Word Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | 43h | Attribute: | WO |
| Default Value: | All bits undefined | Size: | 8 bits |

This register is programmed prior to any counter being accessed to specify counter modes. Following part reset, the control words for each register are undefined and each counter output is 0. Each timer must be programmed to bring it into a known state.

| Bit | Description |
|---|---|
| 7:6 | **Counter Select** — WO. The Counter Selection bits select the counter the control word acts upon as shown below. The Read Back Command is selected when bits[7:6] are both 1.<br>00 = Counter 0 select<br>01 = Counter 1 select<br>10 = Counter 2 select<br>11 = Read Back Command |
| 5:4 | **Read/Write Select** — WO. These bits are the read/write control bits. The actual counter programming is done through the counter port (40h for counter 0, 41h for counter 1, and 42h for counter 2).<br>00 = Counter Latch Command<br>01 = Read/Write Least Significant Byte (LSB)<br>10 = Read/Write Most Significant Byte (MSB)<br>11 = Read/Write LSB then MSB |
| 3:1 | **Counter Mode Selection** — WO. These bits select one of six possible modes of operation for the selected counter.<br>000b — Mode 0 Out signal on end of count (=0)<br>001b — Mode 1 Hardware retriggerable one-shot<br>x10b — Mode 2 Rate generator (divide by n counter)<br>x11b — Mode 3 Square wave output<br>100b — Mode 4 Software triggered strobe<br>101b — Mode 5 Hardware triggered strobe |
| 0 | **Binary/BCD Countdown Select** — WO.<br>0 = Binary countdown is used. The largest possible binary count is $2^{16}$<br>1 = Binary coded decimal (BCD) count is used. The largest possible BCD count is $10^4$ |

There are two special commands that can be issued to the counters through this register, the Read Back Command and the Counter Latch Command. When these commands are chosen, several bits within this register are redefined. These register formats are described below:

### 13.3.2 RDBK_CMD—Read Back Command (LPC I/F—D31:F0)

The Read Back Command is used to determine the count value, programmed mode, and current states of the OUT pin and Null count flag of the selected counter or counters. Status and/or count may be latched in any or all of the counters by selecting the counter during the register write. The count and status remain latched until read, and further latch commands are ignored until the count is read. Both count and status of the selected counters may be latched simultaneously by setting both bit 5 and bit 4 to 0. If both are latched, the first read operation from that counter returns the latched status. The next one or two reads, depending on whether the counter is programmed for one or two byte counts, returns the latched count. Subsequent reads return an unlatched count.

| Bit | Description |
|---|---|
| 7:6 | **Read Back Command.** Must be 11 to select the Read Back Command |
| 5 | **Latch Count of Selected Counters**.<br>0 = Current count value of the selected counters will be latched<br>1 = Current count will not be latched |
| 4 | **Latch Status of Selected Counters**.<br>0 = Status of the selected counters will be latched<br>1 = Status will not be latched |
| 3 | **Counter 2 Select**.<br>1 = Counter 2 count and/or status will be latched |
| 2 | **Counter 1 Select**.<br>1 = Counter 1 count and/or status will be latched |
| 1 | **Counter 0 Select**.<br>1 = Counter 0 count and/or status will be latched. |
| 0 | **Reserved**. Must be 0. |

### 13.3.3 LTCH_CMD—Counter Latch Command (LPC I/F—D31:F0)

The Counter Latch Command latches the current count value. This command is used to insure that the count read from the counter is accurate. The count value is then read from each counter's count register through the Counter Ports Access Ports Register (40h for counter 0, 41h for counter 1, and 42h for counter 2). The count must be read according to the programmed format, i.e., if the counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other (read, write, or programming operations for other counters may be inserted between the reads). If a counter is latched once and then latched again before the count is read, the second Counter Latch Command is ignored.

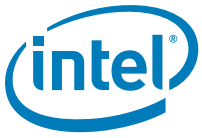
| Bit | Description |
|---|---|
| 7:6 | **Counter Selection.** These bits select the counter for latching. If “11” is written, then the write is interpreted as a read back command.<br>00 = Counter 0<br>01 = Counter 1<br>10 = Counter 2 |
| 5:4 | **Counter Latch Command**.<br>00 = Selects the Counter Latch Command. |
| 3:0 | **Reserved**. Must be 0. |

### 13.3.4 SBYTE_FMT—Interval Timer Status Byte Format Register (LPC I/F—D31:F0)

I/O Address: Counter 0 = 40h, Counter 1 = 41h, Counter 2 = 42h — Attribute:RO — Size:8 bits per counter
Default Value: Bits[6:0] undefined, Bit 7=0

Each counter's status byte can be read following a Read Back Command. If latch status is chosen (bit 4=0, Read Back Command) as a read back option for a given counter, the next read from the counter's Counter Access Ports Register (40h for counter 0, 41h for counter 1, and 42h for counter 2) returns the status byte. The status byte returns the following:

| Bit | Description |
|---|---|
| 7 | **Counter OUT Pin State** — RO.<br>0 = OUT pin of the counter is also a 0<br>1 = OUT pin of the counter is also a 1 |
| 6 | **Count Register Status** — RO. This bit indicates when the last count written to the Count Register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the counter mode, but until the count is loaded into the counting element (CE), the count value will be incorrect.<br>0 = Count has been transferred from CR to CE and is available for reading.<br>1 = Null Count. Count has not been transferred from CR to CE and is not yet available for reading. |
| 5:4 | **Read/Write Selection Status** — RO. These bits reflect the read/write selection made through bits[5:4] of the control register. The binary codes returned during the status read match the codes used to program the counter read/write selection.<br>00 = Counter Latch Command<br>01 = Read/Write Least Significant Byte (LSB)<br>10 = Read/Write Most Significant Byte (MSB)<br>11 = Read/Write LSB then MSB |



| Bit | Description |
|---|---|
| 3:1 | **Mode Selection Status** — RO. These bits return the counter mode programming. The binary code returned matches the code used to program the counter mode, as listed under the bit function above.<br>000 = Mode 0 — Out signal on end of count (=0)<br>001 = Mode 1 — Hardware retriggerable one-shot<br>x10 = Mode 2 — Rate generator (divide by n counter)<br>x11 = Mode 3 — Square wave output<br>100 = Mode 4 — Software triggered strobe<br>101 = Mode 5 — Hardware triggered strobe |
| 0 | **Countdown Type Status** — RO. This bit reflects the current countdown type.<br>0 = Binary countdown<br>1 = Binary Coded Decimal (BCD) countdown. |

### 13.3.5 Counter Access Ports Register (LPC I/F—D31:F0)

I/O Address: Counter 0 – 40h, Counter 1 – 41h, Counter 2 – 42h — Attribute: R/W

Default Value: All bits undefined — Size: 8 bit

| Bit | Description |
|---|---|
| 7:0 | **Counter Port** — R/W. Each counter port address is used to program the 16-bit Count Register. The order of programming, either LSB only, MSB only, or LSB then MSB, is defined with the Interval Counter Control Register at port 43h. The counter port is also used to read the current count from the Count Register, and return the status of the counter programming following a Read Back Command. |

## 13.4 8259 Interrupt Controller (PIC) Registers (LPC I/F—D31:F0)

### 13.4.1 Interrupt Controller I/O MAP (LPC I/F—D31:F0)

The interrupt controller registers are located at 20h and 21h for the master controller (IRQ 0–7), and at A0h and A1h for the slave controller (IRQ 8–13). These registers have multiple functions, depending upon the data written to them. Table 13-121 shows the different register possibilities for each address.

**Table 13-121.PIC Registers (LPC I/F—D31:F0)**

| Port | Aliases | Register Name | Default Value | Type |
|---|---|---|---|---|
| 20h | 24h, 28h, 2Ch, 30h, 34h, 38h, 3Ch | Master PIC ICW1 Init. Cmd Word 1 | Undefined | WO |
| | | Master PIC OCW2 Op Ctrl Word 2 | 001XXXXXb | WO |
| | | Master PIC OCW3 Op Ctrl Word 3 | X01XXX10b | WO |

**Table 13-121.PIC Registers (LPC I/F—D31:F0)**

| Port | Aliases | Register Name | Default Value | Type |
|---|---|---|---|---|
| 21h | 25h, 29h, 2Dh, 31h, 35h, 39h, 3Dh | Master PIC ICW2 Init. Cmd Word 2 | Undefined | WO |
| | | Master PIC ICW3 Init. Cmd Word 3 | Undefined | WO |
| | | Master PIC ICW4 Init. Cmd Word 4 | 01h | WO |
| | | Master PIC OCW1 Op Ctrl Word 1 | 00h | R/W |
| A0h | A4h, A8h, ACh, B0h, B4h, B8h, BCh | Slave PIC ICW1 Init. Cmd Word 1 | Undefined | WO |
| | | Slave PIC OCW2 Op Ctrl Word 2 | 001XXXXXb | WO |
| | | Slave PIC OCW3 Op Ctrl Word 3 | X01XXX10b | WO |
| A1h | A5h, A9h, ADh, B1h, B5h, B9h, BDh | Slave PIC ICW2 Init. Cmd Word 2 | Undefined | WO |
| | | Slave PIC ICW3 Init. Cmd Word 3 | Undefined | WO |
| | | Slave PIC ICW4 Init. Cmd Word 4 | 01h | WO |
| | | Slave PIC OCW1 Op Ctrl Word 1 | 00h | R/W |
| 4D0h | – | Master PIC Edge/Level Triggered | 00h | R/W |
| 4D1h | – | Slave PIC Edge/Level Triggered | 00h | R/W |

***Note:*** Refer to note addressing active-low interrupt sources in 8259 Interrupt Controllers section (Chapter 5.9 - Volume 1).

## 13.4.2 ICW1—Initialization Command Word 1 Register (LPC I/F—D31:F0)

Offset Address: Master Controller – 20h / Slave Controller – A0h
Attribute: WO
Size: 8 bit /controller
Default Value: All bits undefined

A write to Initialization Command Word 1 starts the interrupt controller initialization sequence, during which the following occurs:

1. The Interrupt Mask register is cleared.
2. IRQ7 input is assigned priority 7.
3. The slave mode address is set to 7.
4. Special mask mode is cleared and Status Read is set to IRR.

Once this write occurs, the controller expects writes to ICW2, ICW3, and ICW4 to complete the initialization sequence.

| Bit | Description |
|---|---|
| 7:5 | **ICW/OCW Select** — WO. These bits are MCS-85 specific, and not needed.<br>000 = Should be programmed to “000” |
| 4 | **ICW/OCW Select** — WO.<br>1 = This bit must be a 1 to select ICW1 and enable the ICW2, ICW3, and ICW4 sequence. |

| | |
|---|---|
| 3 | **Edge/Level Bank Select (LTIM)** — WO. Disabled. Replaced by the edge/level triggered control registers (ELCR, D31:F0:4D0h, D31:F0:4D1h). |
| 2 | ADI — WO.<br>0 = Ignored for the Chipset. Should be programmed to 0. |
| 1 | **Single or Cascade (SNGL)** — WO.<br>0 = Must be programmed to a 0 to indicate two controllers operating in cascade mode. |
| 0 | **ICW4 Write Required (IC4)** — WO.<br>1 = This bit must be programmed to a 1 to indicate that ICW4 needs to be programmed. |

### 13.4.3 ICW2—Initialization Command Word 2 Register (LPC I/F—D31:F0)

Offset Address: Master Controller – 21h / Slave Controller – A1h
Attribute: WO
Size: 8 bit /controller
Default Value: All bits undefined

ICW2 is used to initialize the interrupt controller with the five most significant bits of the interrupt vector address. The value programmed for bits[7:3] is used by the processor to define the base address in the interrupt vector table for the interrupt routines associated with each IRQ on the controller. Typical ISA ICW2 values are 08h for the master controller and 70h for the slave controller.

| Bit | Description |
|---|---|
| 7:3 | **Interrupt Vector Base Address** — WO. Bits [7:3] define the base address in the interrupt vector table for the interrupt routines associated with each interrupt request level input. |
| 2:0 | **Interrupt Request Level** — WO. When writing ICW2, these bits should all be 0. During an interrupt acknowledge cycle, these bits are programmed by the interrupt controller with the interrupt to be serviced. This is combined with bits [7:3] to form the interrupt vector driven onto the data bus during the second INTA# cycle. The code is a three bit binary code: |

| Code | Master Interrupt | Slave Interrupt |
|---|---|---|
| 000b | IRQ0 | IRQ8 |
| 001b | IRQ1 | IRQ9 |
| 010b | IRQ2 | IRQ10 |
| 011b | IRQ3 | IRQ11 |
| 100b | IRQ4 | IRQ12 |
| 101b | IRQ5 | IRQ13 |
| 110b | IRQ6 | IRQ14 |
| 111b | IRQ7 | IRQ15 |

### 13.4.4 ICW3—Master Controller Initialization Command Word 3 Register (LPC I/F—D31:F0)

Offset Address: 21h Attribute: WO
Default Value: All bits undefined Size: 8 bits

| Bit | Description |
|---|---|
| 7:3 | 0 = These bits must be programmed to 0. |
| 2 | **Cascaded Interrupt Controller IRQ Connection** — WO. This bit indicates that the slave controller is cascaded on IRQ2. When IRQ8#–IRQ15 is asserted, it goes through the slave controller's priority resolver. The slave controller's INTR output onto IRQ2. IRQ2 then goes through the master controller's priority solver. If it wins, the INTR signal is asserted to the processor, and the returning interrupt acknowledge returns the interrupt vector for the slave controller.<br>1 = This bit must always be programmed to a 1. |
| 1:0 | 0 = These bits must be programmed to 0. |

### 13.4.5 ICW3—Slave Controller Initialization Command Word 3 Register (LPC I/F—D31:F0)

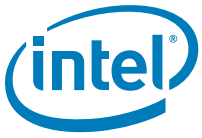
Offset Address: A1h Attribute: WO
Default Value: All bits undefined Size: 8 bits

| Bit | Description |
|---|---|
| 7:3 | 0 = These bits must be programmed to 0. |
| 2:0 | **Slave Identification Code** — WO. These bits are compared against the slave identification code broadcast by the master controller from the trailing edge of the first internal INTA# pulse to the trailing edge of the second internal INTA# pulse. These bits must be programmed to 02h to match the code broadcast by the master controller. When 02h is broadcast by the master controller during the INTA# sequence, the slave controller assumes responsibility for broadcasting the interrupt vector. |

### 13.4.6 ICW4—Initialization Command Word 4 Register (LPC I/F—D31:F0)

Offset Address: Master Controller – 021h Attribute: WO
Slave Controller – 0A1h Size: 8 bits
Default Value: 01h

| Bit | Description |
|---|---|
| 7:5 | 0 = These bits must be programmed to 0. |
| 4 | **Special Fully Nested Mode (SFNM)** — WO.<br>0 = Should normally be disabled by writing a 0 to this bit.<br>1 = Special fully nested mode is programmed. |
| 3 | **Buffered Mode (BUF)** — WO.<br>0 = Must be programmed to 0 for the Chipset. This is non-buffered mode. |



| Bit | Description |
|---|---|
| 2 | **Master/Slave in Buffered Mode** — WO. Not used.<br>0 = Should always be programmed to 0. |
| 1 | **Automatic End of Interrupt (AEOI)** — WO.<br>0 = This bit should normally be programmed to 0. This is the normal end of interrupt.<br>1 = Automatic End of Interrupt (AEOI) mode is programmed. |
| 0 | **Microprocessor Mode** — WO.<br>1 = Must be programmed to 1 to indicate that the controller is operating in an Intel Architecture-based system. |

## 13.4.7 OCW1—Operational Control Word 1 (Interrupt Mask) Register (LPC I/F—D31:F0)

Offset Address: Master Controller – 021h Attribute: R/W
Slave Controller – 0A1h Size: 8 bits
Default Value: 00h

| Bit | Description |
|---|---|
| 7:0 | **Interrupt Request Mask** — R/W. When a 1 is written to any bit in this register, the corresponding IRQ line is masked. When a 0 is written to any bit in this register, the corresponding IRQ mask bit is cleared, and interrupt requests will again be accepted by the controller. Masking IRQ2 on the master controller will also mask the interrupt requests from the slave controller. |

## 13.4.8 OCW2—Operational Control Word 2 Register (LPC I/F—D31:F0)

Offset Address: Master Controller – 020h Attribute: WO
Slave Controller – 0A0h Size: 8 bits
Default Value: Bit[4:0]=undefined, Bit[7:5]=001

Following a part reset or ICW initialization, the controller enters the fully nested mode of operation. Non-specific EOI without rotation is the default. Both rotation mode and specific EOI mode are disabled following initialization.

| Bit | Description |
|---|---|
| 7:5 | **Rotate and EOI Codes** (R, SL, EOI) — WO. These three bits control the Rotate and End of Interrupt modes and combinations of the two.<br>000 = Rotate in Auto EOI Mode (Clear)<br>001 = Non-specific EOI command<br>010 = No Operation<br>011 = *Specific EOI Command<br>100 = Rotate in Auto EOI Mode (Set)<br>101 = Rotate on Non-Specific EOI Command<br>110 = *Set Priority Command<br>111 = *Rotate on Specific EOI Command<br>*L0 – L2 Are Used |
| 4:3 | **OCW2 Select** — WO. When selecting OCW2, bits 4:3 = “00” |

| Bit | Description |
|---|---|
| 2:0 | **Interrupt Level Select** (L2, L1, L0) — WO. L2, L1, and L0 determine the interrupt level acted upon when the SL bit is active. A simple binary code, outlined below, selects the channel for the command to act upon. When the SL bit is inactive, these bits do not have a defined function; programming L2, L1 and L0 to 0 is sufficient in this case. |

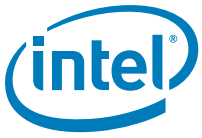
| Code | Interrupt Level |
|---|---|
| 000b | IRQ0/8 |
| 001b | IRQ1/9 |
| 010b | IRQ2/10 |
| 011b | IRQ3/11 |

| Code | Interrupt Level |
|---|---|
| 000b | IRQ4/12 |
| 001b | IRQ5/13 |
| 010b | IRQ6/14 |
| 011b | IRQ7/15 |

## 13.4.9 OCW3—Operational Control Word 3 Register (LPC I/F—D31:F0)

Offset Address: Master Controller – 020h Attribute: WO
Slave Controller – 0A0h Size: 8 bits
Default Value: Bit[6,0]=0, Bit[7,4:2]=undefined, Bit[5,1]=1

| Bit | Description |
|---|---|
| 7 | **Reserved**. Must be 0. |
| 6 | **Special Mask Mode (SMM)** — WO.<br>1 = The Special Mask Mode can be used by an interrupt service routine to dynamically alter the system priority structure while the routine is executing, through selective enabling/disabling of the other channel's mask bits. Bit 5, the ESMM bit, must be set for this bit to have any meaning. |
| 5 | **Enable Special Mask Mode (ESMM)** — WO.<br>0 = Disable. The SMM bit becomes a “don't care”.<br>1 = Enable the SMM bit to set or reset the Special Mask Mode. |
| 4:3 | **OCW3 Select** — WO. When selecting OCW3, bits 4:3 = 01 |
| 2 | **Poll Mode Command** — WO.<br>0 = Disable. Poll Command is not issued.<br>1 = Enable. The next I/O read to the interrupt controller is treated as an interrupt acknowledge cycle. An encoded byte is driven onto the data bus, representing the highest priority level requesting service. |
| 1:0 | **Register Read Command** — WO. These bits provide control for reading the In-Service Register (ISR) and the Interrupt Request Register (IRR). When bit 1=0, bit 0 will not affect the register read selection. When bit 1=1, bit 0 selects the register status returned following an OCW3 read. If bit 0=0, the IRR will be read. If bit 0=1, the ISR will be read. Following ICW initialization, the default OCW3 port address read will be “read IRR”. To retain the current selection (read ISR or read IRR), always write a 0 to bit 1 when programming this register. The selected register can be read repeatedly without reprogramming OCW3. To select a new status register, OCW3 must be reprogrammed prior to attempting the read.<br>00 = No Action<br>01 = No Action<br>10 = Read IRQ Register<br>11 = Read IS Register |



## 13.4.10 ELCR1—Master Controller Edge/Level Triggered Register (LPC I/F—D31:F0)

Offset Address: 4D0h
Default Value: 00h
Attribute: R/W
Size: 8 bits

In edge mode, (bit[x] = 0), the interrupt is recognized by a low to high transition. In level mode (bit[x] = 1), the interrupt is recognized by a high level. The cascade channel, IRQ2, the heart beat timer (IRQ0), and the keyboard controller (IRQ1), cannot be put into level mode.

| Bit | Description |
|---|---|
| 7 | **IRQ7 ECL** — R/W.<br>0 = Edge.<br>1 = Level. |
| 6 | **IRQ6 ECL** — R/W.<br>0 = Edge.<br>1 = Level. |
| 5 | **IRQ5 ECL** — R/W.<br>0 = Edge.<br>1 = Level. |
| 4 | **IRQ4 ECL** — R/W.<br>0 = Edge.<br>1 = Level. |
| 3 | **IRQ3 ECL** — R/W.<br>0 = Edge.<br>1 = Level. |
| 2:0 | **Reserved**. Must be 0. |

## 13.4.11 ELCR2—Slave Controller Edge/Level Triggered Register (LPC I/F—D31:F0)

Offset Address: 4D1h
Default Value: 00h
Attribute: R/W
Size: 8 bits

In edge mode, (bit[x] = 0), the interrupt is recognized by a low to high transition. In level mode (bit[x] = 1), the interrupt is recognized by a high level. The real time clock, IRQ8#, and the floating point error interrupt, IRQ13, cannot be programmed for level mode.

| Bit | Description |
|---|---|
| 7 | **IRQ15 ECL** — R/W.<br>0 = Edge<br>1 = Level |
| 6 | **IRQ14 ECL** — R/W.<br>0 = Edge<br>1 = Level |
| 5 | **Reserved**. Must be 0. |

| Bit | Description |
|---|---|
| 4 | **IRQ12 ECL** — R/W.<br>0 = Edge<br>1 = Level |
| 3 | **IRQ11 ECL** — R/W.<br>0 = Edge<br>1 = Level |
| 2 | **IRQ10 ECL** — R/W.<br>0 = Edge<br>1 = Level |
| 1 | **IRQ9 ECL** — R/W.<br>0 = Edge<br>1 = Level |
| 0 | **Reserved**. Must be 0. |

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

## 13.5.1 APIC Register Map (LPC I/F—D31:F0)

The APIC is accessed via an indirect addressing scheme. Two registers are visible by software for manipulation of most of the APIC registers. These registers are mapped into memory space. The registers are shown in Table 13-122.

**Table 13-122.APIC Direct Registers (LPC I/F—D31:F0)**

| Address | Mnemonic | Register Name | Size | Type |
|---|---|---|---|---|
| FEC0_0000h | IND | Index | 8 bits | R/W |
| FEC0_0010h | DAT | Data | 32 bits | R/W |
| FECO_0040h | EOIR | EOI | 32 bits | WO |

Table 13-123 lists the registers which can be accessed within the APIC via the Index Register. When accessing these registers, accesses must be done one DWord at a time. For example, software should not access byte 2 from the Data register before accessing bytes 0 and 1. The hardware will not attempt to recover from a bad programming model in this case.

**Table 13-123.APIC Indirect Registers (LPC I/F—D31:F0)**

| Index | Mnemonic | Register Name | Size | Type |
|---|---|---|---|---|
| 00 | ID | Identification | 32 bits | R/W |
| 01 | VER | Version | 32 bits | RO |
| 02–0F | — | **Reserved** | — | RO |
| 10–11 | REDIR_TBL0 | Redirection Table 0 | 64 bits | R/W, RO |

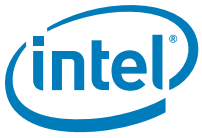


**Table 13-123.APIC Indirect Registers (LPC I/F—D31:F0)**

| Index | Mnemonic | Register Name | Size | Type |
|---|---|---|---|---|
| 12–13 | REDIR_TBL1 | Redirection Table 1 | 64 bits | R/W, RO |
| … | … | … | … | … |
| 3E–3F | REDIR_TBL23 | Redirection Table 23 | 64 bits | R/W, RO |
| 40–FF | — | **Reserved** | — | RO |

## 13.5.2 IND—Index Register (LPC I/F—D31:F0)

Memory Address FEC0_0000h Attribute: R/W
Default Value: 00h Size: 8 bits

The Index Register will select which APIC indirect register to be manipulated by software. The selector values for the indirect registers are listed in Table 13-123. Software will program this register to select the desired APIC internal register

| Bit | Description |
|---|---|
| 7:0 | **APIC Index** — R/W. This is an 8-bit pointer into the I/O APIC register table. |

## 13.5.3 DAT—Data Register (LPC I/F—D31:F0)

Memory Address FEC0_0010h Attribute: R/W
Default Value: 00000000h Size: 32 bits

This is a 32-bit register specifying the data to be read or written to the register pointed to by the Index register. This register can only be accessed in DWord quantities.

| Bit | Description |
|---|---|
| 7:0 | **APIC Data** — R/W. This is a 32-bit register for the data to be read or written to the APIC indirect register (Table 13-123) pointed to by the Index register (Memory Address FEC0_0000h). |

## 13.5.4 EOIR—EOI Register (LPC I/F—D31:F0)

Memory Address FEC0h_0040h Attribute: WO
Default Value: N/A Size: 32 bits

The EOI register is present to provide a mechanism to maintain the level triggered semantics for level-triggered interrupts issued on the parallel bus.

When a write is issued to this register, the I/O APIC will check the lower 8 bits written to this register, and compare it with the vector field for each entry in the I/O Redirection Table. When a match is found, the Remote_IRR bit (Index Offset 10h, bit 14) for that I/O Redirection Entry will be cleared.

***Note:*** If multiple I/O Redirection entries, for any reason, assign the same vector for more than one interrupt input, each of those entries will have the Remote_IRR bit reset to 0. The interrupt which was prematurely reset will not be lost because if its input remained active when the Remote_IRR bit is cleared, the interrupt will be reissued and serviced

at a later time. Note: Only bits 7:0 are actually used. Bits 31:8 are ignored by the Chipset.

***Note:*** To provide for future expansion, the processor should always write a value of 0 to Bits 31:8.

| Bit | Description |
|---|---|
| 31:8 | **Reserved**. To provide for future expansion, the processor should always write a value of 0 to Bits 31:8. |
| 7:0 | **Redirection Entry Clear** — WO. When a write is issued to this register, the I/O APIC will check this field, and compare it with the vector field for each entry in the I/O Redirection Table. When a match is found, the Remote_IRR bit for that I/O Redirection Entry will be cleared. |

### 13.5.5 ID—Identification Register (LPC I/F—D31:F0)

Index Offset: 00h Attribute: R/W
Default Value: 00000000h Size: 32 bits

The APIC ID serves as a physical name of the APIC. The APIC bus arbitration ID for the APIC is derived from its I/O APIC ID. This register is reset to 0 on power-up reset.

| Bit | Description |
|---|---|
| 31:28 | **Reserved** |
| 27:24 | **APIC ID** — R/W. Software must program this value before using the APIC. |
| 23:16 | **Reserved** |
| 15 | Scratchpad Bit. |
| 14:0 | **Reserved** |

### 13.5.6 VER—Version Register (LPC I/F—D31:F0)

Index Offset: 01h Attribute: RO
Default Value: 00170020h Size: 32 bits

Each I/O APIC contains a hardwired Version Register that identifies different implementation of APIC and their versions. The maximum redirection entry information also is in this register, to let software know how many interrupt are supported by this APIC.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** |
| 23:16 | Maximum Redirection Entries — RO. This field is the entry number (0 being the lowest entry) of the highest entry in the redirection table. It is equal to the number of interrupt input pins minus one and is in the range 0 through 239. In the Chipset this field is hardwired to 17h to indicate 24 interrupts. |

| | |
|---|---|
| 15 | PRQ — RO. This bit indicate that the IOxAPIC does not implement the Pin Assertion Register. |
| 14:8 | **Reserved** |
| 7:0 | Version — RO. This is a version number that identifies the implementation version. |

## 13.5.7 REDIR_TBL—Redirection Table (LPC I/F—D31:F0)

Index Offset: 10h–11h (vector 0) throughAttribute: R/W, RO
3E–3Fh (vector 23)
Default Value: Bit 16 = 1,. All other bits undefined Size:64 bits each, (accessed as two 32 bit quantities)

The Redirection Table has a dedicated entry for each interrupt input pin. The information in the Redirection Table is used to translate the interrupt manifestation on the corresponding interrupt pin into an APIC message.

The APIC will respond to an edge triggered interrupt as long as the interrupt is held until after the acknowledge cycle has begun. Once the interrupt is detected, a delivery status bit internally to the I/O APIC is set. The state machine will step ahead and wait for an acknowledgment from the APIC unit that the interrupt message was sent. Only then will the I/O APIC be able to recognize a new edge on that interrupt pin. That new edge will only result in a new invocation of the handler if its acceptance by the destination APIC causes the Interrupt Request Register bit to go from 0 to 1. (In other words, if the interrupt was not already pending at the destination.)

| Bit | Description |
|---|---|
| 63:56 | **Destination** — R/W. If bit 11 of this entry is 0 (Physical), then bits 59:56 specifies an APIC ID. In this case, bits 63:59 should be programmed by software to 0.<br>If bit 11 of this entry is 1 (Logical), then bits 63:56 specify the logical destination address of a set of processors. |
| 55:48 | **Extended Destination ID (EDID)** — RO. These bits are sent to a local APIC only when in Processor System Bus mode. They become bits 11:4 of the address. |
| 47:17 | **Reserved** |
| 16 | **Mask** — R/W.<br>0 = Not masked: An edge or level on this interrupt pin results in the delivery of the interrupt to the destination.<br>1 = Masked: Interrupts are not delivered nor held pending. Setting this bit after the interrupt is accepted by a local APIC has no effect on that interrupt. This behavior is identical to the device withdrawing the interrupt before it is posted to the processor. It is software's responsibility to deal with the case where the mask bit is set after the interrupt message has been accepted by a local APIC unit but before the interrupt is dispensed to the processor. |
| 15 | **Trigger Mode** — R/W. This field indicates the type of signal on the interrupt pin that triggers an interrupt.<br>0 = Edge triggered.<br>1 = Level triggered. |
| 14 | **Remote IRR** — R/W. This bit is used for level triggered interrupts; its meaning is undefined for edge triggered interrupts.<br>0 = Reset when an EOI message is received from a local APIC.<br>1 = Set when Local APIC/s accept the level interrupt sent by the I/O APIC. |

| Bit | Description |
|---|---|
| 13 | **Interrupt Input Pin Polarity** — R/W. This bit specifies the polarity of each interrupt signal connected to the interrupt pins.<br>0 = Active high.<br>1 = Active low. |
| 12 | **Delivery Status** — RO. This field contains the current status of the delivery of this interrupt. Writes to this bit have no effect.<br>0 = Idle. No activity for this interrupt.<br>1 = Pending. Interrupt has been injected, but delivery is not complete. |
| 11 | **Destination Mode** — R/W. This field determines the interpretation of the Destination field.<br>0 = Physical. Destination APIC ID is identified by bits 59:56.<br>1 = Logical. Destinations are identified by matching bit 63:56 with the Logical Destination in the Destination Format Register and Logical Destination Register in each Local APIC. |
| 10:8 | **Delivery Mode** — R/W. This field specifies how the APICs listed in the destination field should act upon reception of this signal. Certain Delivery Modes will only operate as intended when used in conjunction with a specific trigger mode. These encodings are listed in the note below: |
| 7:0 | **Vector** — R/W. This field contains the interrupt vector for this interrupt. Values range between 10h and FEh. |

**NOTE:** Delivery Mode encoding:

000 = Fixed. Deliver the signal on the INTR signal of all processor cores listed in the destination. Trigger Mode can be edge or level.

001 = Lowest Priority. Deliver the signal on the INTR signal of the processor core that is executing at the lowest priority among all the processors listed in the specified destination. Trigger Mode can be edge or level.

010 = SMI (System Management Interrupt). Requires the interrupt to be programmed as edge triggered. The vector information is ignored but must be programmed to all 0's for future compatibility: **not supported**

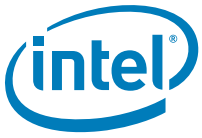
011 = **Reserved**

100 = NMI. Deliver the signal on the NMI signal of all processor cores listed in the destination. Vector information is ignored. NMI is treated as an edge triggered interrupt even if it is programmed as level triggered. For proper operation this redirection table entry must be programmed to edge triggered. The NMI delivery mode does not set the RIRR bit. If the redirection table is incorrectly set to level, the loop count will continue counting through the redirection table addresses. Once the count for the NMI pin is reached again, the interrupt will be sent again: **not supported**

101 = INIT. Deliver the signal to all processor cores listed in the destination by asserting the INIT signal. All addressed local APICs will assume their INIT state. INIT is always treated as an edge triggered interrupt even if programmed as level triggered. For proper operation this redirection table entry must be programmed to edge triggered. The INIT delivery mode does not set the RIRR bit. If the redirection table is incorrectly set to level, the loop count will continue counting through the redirection table addresses. Once the count for the INIT pin is reached again, the interrupt will be sent again: **not supported**

110 = **Reserved**

111 = ExtINT. Deliver the signal to the INTR signal of all processor cores listed in the destination as an interrupt that originated in an externally connected 8259A compatible interrupt controller. The INTA cycle that corresponds to this ExtINT delivery will be routed to the external controller that is expected to supply the vector. Requires the interrupt to be programmed as edge triggered.



# 13.6 Real Time Clock Registers (LPC I/F—D31:F0)

## 13.6.1 I/O Register Address Map (LPC I/F—D31:F0)

The RTC internal registers and RAM are organized as two banks of 128 bytes each, called the standard and extended banks. The first 14 bytes of the standard bank contain the RTC time and date information along with four registers, A–D, that are used for configuration of the RTC. The extended bank contains a full 128 bytes of battery backed SRAM, and will be accessible even when the RTC module is disabled (via the RTC configuration register). Registers A–D do not physically exist in the RAM.

All data movement between the host processor and the real-time clock is done through registers mapped to the standard I/O space. The register map appears in Table 13-124.

**Table 13-124.RTC I/O Registers (LPC I/F—D31:F0)**

| I/O Locations | If U128E bit = 0 | Function |
|---|---|---|
| 70h and 74h | Also alias to 72h and 76h | Real-Time Clock (Standard RAM) Index Register |
| 71h and 75h | Also alias to 73h and 77h | Real-Time Clock (Standard RAM) Target Register |
| 72h and 76h | | Extended RAM Index Register (if enabled) |
| 73h and 77h | | Extended RAM Target Register (if enabled) |

**NOTES:**

1. I/O locations 70h and 71h are the standard legacy location for the real-time clock. The map for this bank is shown in Table 13-125. Locations 72h and 73h are for accessing the extended RAM. The extended RAM bank is also accessed using an indexed scheme. I/O address 72h is used as the address pointer and I/O address 73h is used as the data register. Index addresses above 127h are not valid. If the extended RAM is not needed, it may be disabled.
2. Software must preserve the value of bit 7 at I/O addresses 70h and 74h. When writing to this address, software must first read the value, and then write the same value for bit 7 during the sequential address write. Note that port 70h is not directly readable. The only way to read this register is through Alt Access mode. Although RTC Index bits 6:0 are readable from port 74h, bit 7 will always return 0. If the NMI# enable is not changed during normal operation, software can alternatively read this bit once and then retain the value for all subsequent writes to port 70h.

## 13.6.2 Indexed Registers (LPC I/F—D31:F0)

The RTC contains two sets of indexed registers that are accessed using the two separate Index and Target registers (70/71h or 72/73h), as shown in Table 13-125.

**Table 13-125.RTC (Standard) RAM Bank (LPC I/F—D31:F0)**

| Index | Name |
|---|---|
| 00h | Seconds |
| 01h | Seconds Alarm |
| 02h | Minutes |
| 03h | Minutes Alarm |
| 04h | Hours |
| 05h | Hours Alarm |
| 06h | Day of Week |
| 07h | Day of Month |
| 08h | Month |
| 09h | Year |
| 0Ah | Register A |
| 0Bh | Register B |
| 0Ch | Register C |
| 0Dh | Register D |
| 0Eh–7Fh | 114 Bytes of User RAM |

### 13.6.2.1 RTC_REGA—Register A (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| RTC Index: | 0A | Attribute: | R/W |
| Default Value: | Undefined | Size: | 8-bit |
| Lockable: | No | Power Well: | RTC |

This register is used for general configuration of the RTC functions. None of the bits are affected by RSMRST# or any other Chipset reset signal.

| Bit | Description |
|---|---|
| 7 | **Update In Progress (UIP)** — R/W. This bit may be monitored as a status flag.<br>0 = The update cycle will not start for at least 488 µs. The time, calendar, and alarm information in RAM is always available when the UIP bit is 0.<br>1 = The update is soon to occur or is in progress. |
| 6:4 | **Division Chain Select (DV[2:0])** — R/W. These three bits control the divider chain for the oscillator, and are not affected by RSMRST# or any other reset signal. DV2 corresponds to bit 6.<br>010 = Normal Operation<br>11X = Divider Reset<br>101 = Bypass 15 stages (test mode only)<br>100 = Bypass 10 stages (test mode only)<br>011 = Bypass 5 stages (test mode only)<br>001 = Invalid<br>000 = Invalid |

| Bit | Description |
|---|---|
| 3:0 | **Rate Select (RS[3:0])** — R/W. Selects one of 13 taps of the 15 stage divider chain. The selected tap can generate a periodic interrupt if the PIE bit is set in Register B. Otherwise this tap will set the PF flag of Register C. If the periodic interrupt is not to be used, these bits should all be set to 0. RS3 corresponds to bit 3.<br>0000 = Interrupt does not toggle<br>0001 = 3.90625 ms<br>0010 = 7.8125 ms<br>0011 = 122.070 µs<br>0100 = 244.141 µs<br>0101 = 488.281 µs<br>0110 = 976.5625 µs<br>0111 = 1.953125 ms<br>1000 = 3.90625 ms<br>1001 = 7.8125 ms<br>1010 = 15.625 ms<br>1011 = 31.25 ms<br>1100 = 62.5 ms<br>1101 = 125 ms<br>1110 = 250 ms<br>1111= 500 ms |

### 13.6.2.2 RTC_REGB—Register B (General Configuration) (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| RTC Index: | 0Bh | Attribute: | R/W |
| Default Value: | U0U00UUU (U: Undefined) | Size: | 8-bit |
| Lockable: | No | Power Well: | RTC |

| Bit | Description |
|---|---|
| 7 | **Update Cycle Inhibit (SET)** — R/W. Enables/Inhibits the update cycles. This bit is not affected by RSMRST# nor any other reset signal.<br>0 = Update cycle occurs normally once each second.<br>1 = A current update cycle will abort and subsequent update cycles will not occur until SET is returned to 0. When set is one, the BIOS may initialize time and calendar bytes safely.<br><br>**NOTE:** This bit should be set then cleared early in BIOS POST after each powerup directly after coin-cell battery insertion. |
| 6 | **Periodic Interrupt Enable (PIE)** — R/W. This bit is cleared by RSMRST#, but not on any other reset.<br>0 = Disable.<br>1 = Enable. Allows an interrupt to occur with a time base set with the RS bits of register A. |
| 5 | **Alarm Interrupt Enable (AIE)** — R/W. This bit is cleared by RTCRST#, but not on any other reset.<br>0 = Disable.<br>1 = Enable. Allows an interrupt to occur when the AF is set by an alarm match from the update cycle. An alarm can occur once a second, one an hour, once a day, or one a month. |

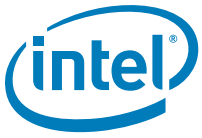
| Bit | Description |
|---|---|
| 4 | **Update-Ended Interrupt Enable (UIE)** — R/W. This bit is cleared by RSMRST#, but not on any other reset.<br>0 = Disable.<br>1 = Enable. Allows an interrupt to occur when the update cycle ends. |
| 3 | **Square Wave Enable (SQWE)** — R/W. This bit serves no function in the Chipset. It is left in this register bank to provide compatibility with the Motorola 146818B. The Chipset has no SQW pin. This bit is cleared by RSMRST#, but not on any other reset. |
| 2 | **Data Mode (DM)** — R/W. This bit specifies either binary or BCD data representation. This bit is not affected by RSMRST# nor any other reset signal.<br>0 = BCD<br>1 = Binary |
| 1 | **Hour Format (HOURFORM)** — R/W. This bit indicates the hour byte format. This bit is not affected by RSMRST# nor any other reset signal.<br>0 = Twelve-hour mode. In twelve-hour mode, the seventh bit represents AM as 0 and PM as one.<br>1 = Twenty-four hour mode. |
| 0 | **Daylight Savings Enable (DSE)** — R/W. This bit triggers two special hour updates per year. The days for the hour adjustment are those specified in United States federal law as of 1987, which is different than previous years. This bit is not affected by RSMRST# nor any other reset signal.<br>0 = Daylight Savings Time updates do not occur.<br>1 = a) Update on the first Sunday in April, where time increments from 1:59:59 AM to 3:00:00 AM.<br>b) Update on the last Sunday in October when the time first reaches 1:59:59 AM, it is changed to 1:00:00 AM. The time must increment normally for at least two update cycles (seconds) previous to these conditions for the time change to occur properly. |

### 13.6.2.3 RTC_REGC—Register C (Flag Register) (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| RTC Index: | 0Ch | Attribute: | RO |
| Default Value: | 00U00000 (U: Undefined) | Size: | 8-bit |
| Lockable: | No | Power Well: | RTC |

Writes to Register C have no effect.

| Bit | Description |
|---|---|
| 7 | **Interrupt Request Flag (IRQF)** — RO. IRQF = (PF * PIE) + (AF * AIE) + (UF *UFE). This bit also causes the RTC Interrupt to be asserted. This bit is cleared upon RSMRST# or a read of Register C. |
| 6 | **Periodic Interrupt Flag (PF)** — RO. This bit is cleared upon RSMRST# or a read of Register C.<br>0 = If no taps are specified via the RS bits in Register A, this flag will not be set.<br>1 = Periodic interrupt Flag will be 1 when the tap specified by the RS bits of register A is 1. |



| Bit | Description |
|---|---|
| 5 | **Alarm Flag (AF)** — RO.<br>0 = This bit is cleared upon RTCRST# or a read of Register C.<br>1 = Alarm Flag will be set after all Alarm values match the current time. |
| 4 | **Update-Ended Flag (UF)** — RO.<br>0 = The bit is cleared upon RSMRST# or a read of Register C.<br>1 = Set immediately following an update cycle for each second. |
| 3:0 | **Reserved**. Will always report 0. |

### 13.6.2.4 RTC_REGD—Register D (Flag Register) (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| RTC Index: | 0Dh | Attribute: | R/W |
| Default Value: | 10UUUUUU (U: Undefined) | Size: | 8-bit |
| Lockable: | No | Power Well: | RTC |

| Bit | Description |
|---|---|
| 7 | **Valid RAM and Time Bit (VRT)** — R/W.<br>0 = This bit should always be written as a 0 for write cycle, however it will return a 1 for read cycles.<br>1 = This bit is hardwired to 1 in the RTC power well. |
| 6 | **Reserved**. This bit always returns a 0 and should be set to 0 for write cycles. |
| 5:0 | **Date Alarm** — R/W. These bits store the date of month alarm value. If set to 000000b, then a don't care state is assumed. The host must configure the date alarm for these bits to do anything, yet they can be written at any time. If the date alarm is not enabled, these bits will return 0's to mimic the functionality of the Motorola 146818B. These bits are not affected by any reset assertion. |

## 13.7 Processor Interface Registers (LPC I/F—D31:F0)

Table 13-126 is the register address map for the processor interface registers.

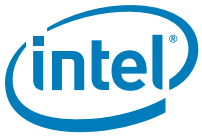
**Table 13-126.Processor Interface PCI Register Address Map (LPC I/F—D31:F0)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 61h | NMI_SC | NMI Status and Control | 00h | R/W, RO |
| 70h | NMI_EN | NMI Enable | 80h | R/W (special) |
| 92h | PORT92 | Fast A20 and Init | 00h | R/W |
| F0h | COPROC_ERR | Coprocessor Error | 00h | R/W |
| CF9h | RST_CNT | Reset Control | 00h | R/W |

### 13.7.1 NMI_SC—NMI Status and Control Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | 61h | Attribute: | R/W, RO |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7 | **SERR# NMI Source Status (SERR#_NMI_STS)** — RO.<br>1 = Bit is set if a PCI agent detected a system error and pulses the PCI SERR# line and if bit 2 (PCI_SERR_EN) is cleared. This interrupt source is enabled by setting bit 2 to 0. To reset the interrupt, set bit 2 to 1 and then set it to 0. When writing to port 61h, this bit must be 0.<br><br>**NOTE:** This bit is set by any of the Chipset internal sources of SERR; this includes SERR assertions forwarded from the secondary PCI bus, errors on a PCI Express* port, or other internal functions that generate SERR#. |
| 6 | **IOCHK# NMI Source Status (IOCHK_NMI_STS)** — RO.<br>1 = Bit is set if an LPC agent (via SERIRQ) asserted IOCHK# and if bit 3 (IOCHK_NMI_EN) is cleared. This interrupt source is enabled by setting bit 3 to 0. To reset the interrupt, set bit 3 to 1 and then set it to 0. When writing to port 61h, this bit must be a 0. |
| 5 | **Timer Counter 2 OUT Status (TMR2_OUT_STS)** — RO. This bit reflects the current state of the 8254 counter 2 output. Counter 2 must be programmed following any PCI reset for this bit to have a determinate value. When writing to port 61h, this bit must be a 0. |
| 4 | **Refresh Cycle Toggle (REF_TOGGLE)** — RO. This signal toggles from either 0 to 1 or 1 to 0 at a rate that is equivalent to when refresh cycles would occur. When writing to port 61h, this bit must be a 0. |
| 3 | **IOCHK# NMI Enable (IOCHK_NMI_EN)** — R/W.<br>0 = Enabled.<br>1 = Disabled and cleared. |
| 2 | **PCI SERR# Enable (PCI_SERR_EN)** — R/W.<br>0 = SERR# NMIs are enabled.<br>1 = SERR# NMIs are disabled and cleared. |
| 1 | **Speaker Data Enable (SPKR_DAT_EN)** — R/W.<br>0 = SPKR output is a 0.<br>1 = SPKR output is equivalent to the Counter 2 OUT signal value. |
| 0 | **Timer Counter 2 Enable (TIM_CNT2_EN)** — R/W.<br>0 = Disable<br>1 = Enable |



### 13.7.2 NMI_EN—NMI Enable (and Real Time Clock Index) Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | 70h | Attribute: | R/W (special) |
| Default Value: | 80h | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

***Note:*** The RTC Index field is write-only for normal operation. This field can only be read in Alt-Access Mode. Note, however, that this register is aliased to Port 74h (documented in), and all bits are readable at that address.

| Bits | Description |
|---|---|
| 7 | **NMI Enable (NMI_EN)** — R/W (special).<br>0 = Enable NMI sources.<br>1 = Disable All NMI sources. |
| 6:0 | **Real Time Clock Index Address (RTC_INDX)** — R/W (special). This data goes to the RTC to select which register or CMOS RAM address is being accessed. |

### 13.7.3 PORT92—Fast A20 and Init Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | 92h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:2 | **Reserved** |
| 1 | **Alternate A20 Gate (ALT_A20_GATE)** — R/W. This bit is Or'd with the A20GATE input signal to generate A20M# to the processor.<br>0 = A20M# signal can potentially go active.<br>1 = This bit is set when INIT# goes active. |
| 0 | **INIT_NOW** — R/W. When this bit transitions from a 0 to a 1, the Chipset will force INIT# active for 16 PCI clocks. |

### 13.7.4 COPROC_ERR—Coprocessor Error Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | F0h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8-bits |
| Lockable: | No | Power Well: | Core |

| Bits | Description |
|---|---|
| 7:0 | **Coprocessor Error (COPROC_ERR)** — R/W. Any value written to this register will cause IGNNE# to go active, if FERR# had generated an internal IRQ13. For FERR# to generate an internal IRQ13, the COPROC_ERR_EN bit (Device 31:Function 0, Offset D0, Bit 13) must be 1. Reads to this register always return 00h. |

### 13.7.5 RST_CNT—Reset Control Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | CF9h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:4 | **Reserved** |
| 3 | **Full Reset (FULL_RST)** — R/W. This bit is used to determine the states of SLP_S3#, SLP_S4#, and SLP_S5# after a CF9 hard reset (SYS_RST =1 and RST_CPU is set to 1), after PWROK going low (with RSMRST# high), or after two TCO timeouts.<br>0 = Chipset will keep SLP_S3#, SLP_S4# and SLP_S5# high.<br>1 = Chipset will drive SLP_S3#, SLP_S4# and SLP_S5# low for 3 – 5 seconds.<br><br>**NOTE:** When this bit is set, it also causes the full power cycle (SLP_S3/4/5# assertion) in response to SYS_RESET#, PWROK#, and Watchdog timer reset sources. |
| 2 | **Reset CPU (RST_CPU)** — R/W. When this bit transitions from a 0 to a 1, it initiates a hard or soft reset, as determined by the SYS_RST bit (bit 1 of this register). |
| 1 | **System Reset (SYS_RST)** — R/W. This bit is used to determine a hard or soft reset to the processor.<br>0 = When RST_CPU bit goes from 0 to 1, the Chipset performs a soft reset by activating INIT# for 16 PCI clocks.<br>1 = When RST_CPU bit goes from 0 to 1, the Chipset performs a hard reset by activating PLTRST# and SUS_STAT# active for about 5-6 milliseconds, however the SLP_S3#, SLP_S4# and SLP_S5# will NOT go active. The Chipset main power well is reset when this bit is 1. It also resets the resume well bits (except for those noted throughout the datasheet). |
| 0 | **Reserved** |

## 13.8 Power Management Registers (PM—D31:F0)

The power management registers are distributed within the PCI Device 31: Function 0 space, as well as a separate I/O range. Each register is described below. Unless otherwise indicate, bits are in the main (core) power well.

Bits not explicitly defined in each register are assumed to be reserved. When writing to a reserved bit, the value should always be 0. Software should not attempt to use the value read from a reserved bit, as it may not be consistently 1 or 0.

### 13.8.1 Power Management PCI Configuration Registers (PM—D31:F0)

Table 13-127 shows a small part of the configuration space for PCI Device 31: Function 0. It includes only those registers dedicated for power management. Some of the registers are only used for Legacy Power management schemes.

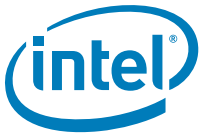


**Table 13-127.Power Management PCI Register Address Map (PM—D31:F0)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| A0h | GEN_PMCON_1 | General Power Management Configuration 1 | 0000h | R/W, RO, R/WO |
| A2h | GEN_PMCON_2 | General Power Management Configuration 2 | 00h | R/W, R/WC |
| A4h | GEN_PMCON_3 | General Power Management Configuration 3 | 00h | R/W, R/WC |
| A9h | Cx-STATE_CNF | Cx State Configuration (Netbook Only). | 00h | R/W |
| AAh | C4-TIMING_CNT | C4 Timing Control (Netbook Only). | 00h | R/W |
| ABh | BM_BREAK_EN | BM_BREAK_EN | 00h | R/W |
| ADh | MSC_FUN | Miscellaneous Functionality | 00h | R/W |
| B8–BBh | GPI_ROUT | GPI Route Control | 00000000h | R/W |

### 13.8.1.1 GEN_PMCON_1—General PM Configuration 1 Register (PM—D31:F0)

Offset Address: A0h
Default Value: 0000h
Lockable: No
Attribute: R/W, RO, R/WO
Size: 16-bit
Usage: ACPI, Legacy
Power Well: Core

| Bit | Description |
|---|---|
| 15:11 | **Reserved** |
| 10 | **BIOS_PCI_EXP_EN** — R/W. This bit acts as a global enable for the SCI associated with the PCI Express* ports.<br>0 = The various PCI Express ports and (G)MCH/CPU cannot cause the PCI_EXP_STS bit to go active.<br>1 = The various PCI Express ports and (G)MCH/CPU can cause the PCI_EXP_STS bit to go active. |
| 9 | **PWRBTN_LVL** — RO. This bit indicates the current state of the PWRBTN# signal.<br>0 = Low.<br>1 = High. |
| 8 | **Reserved** |
| 7 (Nettop Only) | **Reserved** |
| 7 (Netbook Only) | **Enter C4 When C3 Invoked (C4onC3_EN)** — R/W. If this bit is set, then when software does a LVL3 read, the Chipset-M/Chipset-U transitions to the C4 state. |
| 6 | **i64_EN**. Software sets this bit to indicate that the processor is an IA_64 processor, not an IA_32 processor. This may be used in various state machines where there are behavioral differences. |

| Bit | Description |
|---|---|
| 5 (Nettop Only) | **CPU SLP# Enable (CPUSLP_EN)** — R/W.<br>0 = Disable.<br>1 = Enables the CPUSLP# signal to go active in the S1 state. This reduces the processor power. |
| 5 (Netbook Only) | **Reserved** |
| 4 | **SMI_LOCK** — R/WO. When this bit is set, writes to the GLB_SMI_EN bit (PMBASE + 30h, bit 0) will have no effect. Once the SMI_LOCK bit is set, writes of 0 to SMI_LOCK bit will have no effect (i.e., once set, this bit can only be cleared by PLTRST#). |
| 3:2 (Nettop Only) | **Reserved** |
| 3 (Netbook Only) | **Intel SpeedStep Enable (SS_EN)** — R/W.<br>0 = Intel SpeedStep technology logic is disabled and the SS_CNT register will not be visible (reads to SS_CNT will return 00h and writes will have no effect).<br>1 = Intel SpeedStep technology logic is enabled. |
| 2 (Netbook Only) | **PCI CLKRUN# Enable (CLKRUN_EN)** — R/W.<br>0 = Disable. Chipset-M/Chipset-U drives the CLKRUN# signal low.<br>1 = Enable CLKRUN# logic to control the system PCI clock via the CLKRUN# and STP_PCI# signals.<br>**NOTE:** when the SLP_EN# bit is set, the Chipset drives the CLKRUN# signal low regardless of the state of the CLKRUN_EN bit. This ensures that the PCI and LPC clocks continue running during a transition to a sleep state. |
| 1:0 | **Periodic SMI# Rate Select (PER_SMI_SEL)** — R/W. Set by software to control the rate at which periodic SMI# is generated.<br>00 = 1 minute<br>01 = 32 seconds<br>10 = 16 seconds<br>11 = 8 seconds |

### 13.8.1.2 GEN_PMCON_2—General PM Configuration 2 Register (PM—D31:F0)

Offset Address: A2h
Default Value: 00h
Lockable: No
Attribute: R/W, R/WC
Size: 8-bit
Usage: ACPI, Legacy
Power Well: Resume

| Bit | Description |
|---|---|
| 7 | DRAM Initialization Bit — R/W. This bit does not effect hardware functionality in any way. BIOS is expected to set this bit prior to starting the DRAM initialization sequence and to clear this bit after completing the DRAM initialization sequence. BIOS can detect that a DRAM initialization sequence was interrupted by a reset by reading this bit during the boot sequence.<br>• If the bit is 1, then the DRAM initialization was interrupted.<br>• This bit is reset by the assertion of the RSMRST# pin. |

| Bit | Description |
|---|---|
| 6:5 | **CPU PLL Lock Time (CPLT)** — R/W. This field indicates the amount of time that the processor needs to lock its PLLs. This is used wherever timing t270 (Chapter 20) applies.<br>00 = min 30.7 µs (Default)<br>01 = min 61.4 µs<br>10 = min 122.8 µs<br>11 = min 245.6 µs<br>It is the responsibility of the BIOS to program the correct value in this field prior to the first transition to C3 or C4 states (or performing Intel SpeedStep® technology transitions).<br>**NOTE:** The new DPSLP-TO-SLP bits (D31:FO:AAh, bits 1:0) act as an override to these bits.<br>**NOTE:** These bits are not cleared by any type of reset except RSMRST# or a CF9 write |
| 4 | **System Reset Status (SRS)** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = SYS_RESET# button Not pressed.<br>1 = Chipset sets this bit when the SYS_RESET# button is pressed. BIOS is expected to read this bit and clear it, if it is set.<br>**NOTE:** This bit is also reset by RSMRST# and CF9h resets. |
| 3 | **CPU Thermal Trip Status (CTS)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit is set when PLTRST# is inactive and THRMTRIP# goes active while the system is in an S0 or S1 state.<br><br>**NOTES:**<br>1. This bit is also reset by RSMRST#, and CF9h resets. It is not reset by the shutdown and reboot associated with the CPUTHRMTRIP# event.<br>2. The CF9h reset in the description refers to CF9h type core well reset which includes SYS_RST#, PWROK/VRMPWRGD low, SMBus hard reset, TCO Timeout. This type of reset will clear CTS bit. |
| 2 | **Minimum SLP_S4# Assertion Width Violation Status** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Hardware sets this bit when the SLP_S4# assertion width is less than the time programmed in the SLP_S4# Minimum Assertion Width field (D31:F0:Offset A4h:bits 5:4). The Chipset begins the timer when SLP_S4# is asserted during S4/S5 entry, or when the RSMRST# input is deasserted during G3 exit. Note that this bit is functional regardless of the value in the SLP_S4# Assertion Stretch Enable (D31:F0:Offset A4h:bit 3).<br><br>**NOTE:** This bit is reset by the assertion of the RSMRST# pin, but can be set in some cases before the default value is readable. |
| 1 | **CPU Power Failure (CPUPWR_FLR)** — R/WC.<br>0 = Software (typically BIOS) clears this bit by writing a 0 to it.<br>1 = Indicates that the VRMPWRGD signal from the processor's VRM went low while the system was in an S0 or S1 state.<br>**NOTE:** VRMPWRGD is sampled using the RTC clock. Therefore, low times that are less than one RTC clock period may not be detected by the Chipset. |

| Bit | Description |
|---|---|
| 0 | **PWROK Failure (PWROK_FLR)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it, or when the system goes into a G3 state.<br>1 = This bit will be set any time PWROK goes low, when the system was in S0, or S1 state. The bit will be cleared only by software by writing a 1 to this bit or when the system goes to a G3 state.<br>**NOTE:** See Chapter 5.14.11.3 - Volume 1 for more details about the PWROK pin functionality.<br>**NOTE:** In the case of true PWROK failure, PWROK will go low first before the VRMPWRGD. |

**NOTE:** VRMPWROK is sampled using the RTC clock. Therefore, low times that are less than one RTC clock period may not be detected by the Chipset.

### 13.8.1.3 GEN_PMCON_3—General PM Configuration 3 Register (PM—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | A4h | Attribute: | R/W, R/WC |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Usage: | ACPI, Legacy |
| | | Power Well: | RTC |

| Bit | Description |
|---|---|
| 7:6 | **SWSMI_RATE_SEL** — R/W. This field indicates when the SWSMI timer will time out.<br>Valid values are:<br>00 = 1.5 ms ± 0.6 ms<br>01 = 16 ms ± 4 ms<br>10 = 32 ms ± 4 ms<br>11 = 64 ms ± 4 ms<br>These bits are not cleared by any type of reset except RTCRST#. |
| 5:4 | SLP_S4# Minimum Assertion Width — R/W. This field indicates the minimum assertion width of the SLP_S4# signal to ensure that the DRAMs have been safely power-cycled.<br>Valid values are:<br>11 = 1 to 2 seconds<br>10 = 2 to 3 seconds<br>01 = 3 to 4 seconds<br>00 = 4 to 5 seconds<br>This value is used in two ways:<br>1. If the SLP_S4# assertion width is ever shorter than this time, a status bit is set for BIOS to read when S0 is entered.<br>2. If enabled by bit 3 in this register, the hardware will prevent the SLP_S4# signal from deasserting within this minimum time period after asserting.<br>RTCRST# forces this field to the conservative default state (00b) |
| 3 | SLP_S4# Assertion Stretch Enable — R/W.<br>0 = The SLP_S4# minimum assertion time is 1 to 2 RTCCLK.<br>1 = The SLP_S4# signal minimally assert for the time specified in bits 5:4 of this register.<br>This bit is cleared by RTCRST# |
| 2 | **RTC Power Status (RTC_PWR_STS)** — R/W. This bit is set when RTCRST# indicates a weak or missing battery. The bit is not cleared by any type of reset. The bit will remain set until the software clears it by writing a 0 back to this bit position. |

| Bit | Description |
|---|---|
| 1 | **Power Failure (PWR_FLR)** — R/WC. This bit is in the RTC well, and is not cleared by any type of reset except RTCRST#.<br>0 = Indicates that the trickle current has not failed since the last time the bit was cleared. Software clears this bit by writing a 1 to it.<br>1 = Indicates that the trickle current (from the main battery or trickle supply) was removed or failed.<br><br>**NOTE:** Clearing CMOS in an ICH-based platform can be done by using a jumper on RTCRST# or GPI, or using SAFEMODE strap. Implementations should not attempt to clear CMOS by using a jumper to pull VccRTC low. |
| 0 | **AFTERG3_EN** — R/W. This bit determines what state to go to when power is re-applied after a power failure (G3 state). This bit is in the RTC well and is not cleared by any type of reset except writes to CF9h or RTCRST#.<br>0 = System will return to S0 state (boot) after power is re-applied.<br>1 = System will return to the S5 state (except if it was in S4, in which case it will return to S4). In the S5 state, the only enabled wake event is the Power Button or any enabled wake event that was preserved through the power failure.<br><br>**NOTE:** Bit will be set when THRMTRIP#-based shutdown occurs. |

**NOTE:** RSMRST# is sampled using the RTC clock. Therefore, low times that are less than one RTC clock period may not be detected by the Chipset.

### 13.8.1.4 Cx-STATE_CNF—Cx State Configuration Register (PM—D31:F0) (Netbook Only)

| | | | |
|---|---|---|---|
| Offset Address: | A9h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Usage: | ACPI, Legacy |
| Power Well: | Core | | |

This register is used to enable new C-state related modes.

| Bit | Description |
|---|---|
| 7 | **SCRATCHPAD (SP)** — R/W. |
| 6:5 | **Reserved** |
| 4 | **Popdown Mode Enable (PDME)** — R/W. This bit is used in conjunction with the PUME bit (D31:F0:A9h, bit 3). If PUME is 0, then this bit must also be 0.<br>0 = The Chipset on Netbook platform will not attempt to automatically return to a previous C3 or C4 state.<br>1 = When this bit is a 1 and Chipset on Netbook platform observes that there are no bus master requests, it can return to a previous C3 or C4 state.<br><br>**NOTE:** This bit is separate from the PUME bit to cover cases where latency issues permit POPUP but not POPDOWN. |

| Bit | Description |
|---|---|
| 3 | **Popup Mode Enable (PUME)** — R/W. When this bit is a 0, the Chipset on Netbook platform behaves like Chipset, in that bus master traffic is a break event, and it will return from C3/C4 to C0 based on a break event. See Chapter 5.14.5 for additional details on this mode.<br>0 = The Chipset will treat Bus master traffic a break event, and will return from C3/C4 to C0 based on a break event.<br>1 = When this bit is a 1 and Chipset observes a bus master request, it will take the system from a C3 or C4 state to a C2 state and auto enable bus masters. This will let snoops and memory access occur. |
| 2 | **Report Zero for BM_STS (BM_STS_ZERO_EN)** — R/W.<br>0 = The Chipset sets BM_STS (PMBASE + 00h, bit 4) if there is bus master activity from PCI, PCI Express* and internal bus masters.<br>1 = When this bit is a 1, Chipset will not set the BM_STS if there is bus master activity from PCI, PCI Express and internal bus masters.<br><br>**NOTES:**<br>1. If the BM_STS bit is already set when the BM_STS_ZERO_EN bit is set, the BM_STS bit will remain set. Software will still need to clear the BM_STS bit.<br>2. It is expected that if the PUME bit (this register, bit 3) is set, the BM_STS_ZERO_EN bit should also be set. Setting one without the other would mainly be for debug or errata workaround.<br>3. BM_STS will be set by LPC DMA (Netbook Only) or LPC masters, even if BM_STS_ZERO_EN is set. |
| 1:0 | **Reserved** |

### 13.8.1.5 C4-TIMING_CNT—C4 Timing Control Register (PM—D31:F0) (Netbook Only)

| | | | |
|---|---|---|---|
| Offset Address: | AAh | Attribute: | R/W |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Usage: | ACPI, Legacy |
| Power Well: | Core | | |

This register is used to enable C-state related modes.

| Bit | Description |
|---|---|
| 7:4 | **Reserved** |
| 3:2 | **DPRSLPVR to STPCPU** — R/W. This field selects the amount of time that the Chipset on Netbook platform waits for from the deassertion of DPRSLPVR to the deassertion of STP_CPU#. This provides a programmable time for the processor's voltage to stabilize when exiting from a C4 state. This changes the value for t266.<br><br>**Bits** / **$t266_{min}$** / **$t266_{max}$** / **Comment**<br>00b / 95 µs / 101 µs / Default<br>01b / 22 µs / 28 µs / Value used for "Fast" VRMs<br>10b / 34 µs / 40 µs / Recommended Value<br>11b / / / **Reserved** |

| Bit | Description |
|---|---|
| 1:0 | **DPSLP-TO-SLP** — R/W. This field selects the DPSLP# deassertion to CPU_SLP# deassertion time (t270). Normally this value is determined by the CPU_PLL_LOCK_TIME field in the GEN_PMCON_2 register. When this field is non-zero, then the values in this register have higher priority. It is software's responsibility to program these fields in a consistent manner. |

| Bits | t270 |
|---|---|
| 00b | Use value in CPU_PLL_LOCK_TIME field (default is 30 µs) |
| 01b | 20 µs |
| 10b | 15 µs (Recommended Value) |
| 11b | 10 µs |

### 13.8.1.6 BM_BREAK_EN Register (PM—D31:F0) (Netbook Only)

Offset Address: ABh
Default Value: 00h
Lockable: No
Power Well: Core
Attribute: R/W
Size: 8-bit
Usage: ACPI, Legacy

| Bit | Description |
|---|---|
| 7 | **IDE_BREAK_EN** — R/W.<br>0 = Serial ATA traffic will **not** act as a break event.<br>1 = Serial ATA traffic acts as a break event, even if the BM_STS-ZERO_EN and POPUP_EN bits are set. Serial ATA master activity will cause BM_STS to be set and will cause a break from C3/C4. |
| 6 (Netbook Only) | **PCIE_BREAK_EN** — R/W.<br>0 = PCI Express* traffic will **not** act as a break event.<br>1 = PCI Express traffic acts as a break event, even if the BM_STS-ZERO_EN and POPUP_EN bits are set. PCI Express master activity will cause BM_STS to be set and will cause a break from C3/C4. |
| 5 | **PCI_BREAK_EN** — R/W.<br>0 = PCI traffic will **not** act as a break event.<br>1 = PCI traffic acts as a break event, even if the BM_STS-ZERO_EN and POPUP_EN bits are set. PCI master activity will cause BM_STS to be set and will cause a break from C3/C4. |
| 4:3 | **Reserved** |
| 2 | **EHCI_BREAK_EN** — R/W.<br>0 = EHCI traffic will **not** act as a break event.<br>1 = EHCI traffic acts as a break event, even if the BM_STS-ZERO_EN and POPUP_EN bits are set. EHCI master activity will cause BM_STS to be set and will cause a break from C3/C4. |
| 1 | **UHCI_BREAK_EN** — R/W.<br>0 = UHCI traffic will **not** act as a break event.<br>1 = USB traffic from any of the internal UHCIs acts as a break event, even if the BM_STS-ZERO_EN and POPUP_EN bits are set. UHCI master activity will cause BM_STS to be set and will cause a break from C3/C4. |

| Bit | Description |
|---|---|
| 0 | ACAZ_BREAK_EN — R/W.<br>0 = Intel HD Audio traffic will not act as a break event.<br>1 = Intel High Definition Audio traffic acts as a break event, even if the BM_STS-ZERO_EN and POPUP_EN bits are set. Intel High Definition Audio master activity will cause BM_STS to be set and will cause a break from C3/C4. |

### 13.8.1.7 MSC_FUN—Miscellaneous Functionality Register (PM—D31:F0)

Offset Address: ADh
Default Value: 00h
Power Well: Resume
Attribute: R/W
Size: 8-bit

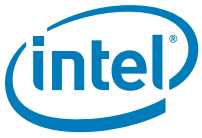
| Bit | Description |
|---|---|
| 7:2 | **Reserved** |
| 1:0 | USB **Transient Disconnect Detect (TDD)** — R/W: This field prevents a short Single-Ended Zero (SE0) condition on the USB ports from being interpreted by the UHCI host controller as a disconnect. BIOS should set to 11b. |

### 13.8.1.8 GPIO_ROUT—GPIO Routing Control Register (PM—D31:F0)

Offset Address: B8h – BBh
Default Value: 00000000h
Lockable: No
Attribute: R/W
Size: 32-bit
Power Well: Resume

| Bit | Description |
|---|---|
| 31:30 | **GPIO15 Route** — R/W. See bits 1:0 for description. |
| Same pattern for GPIO14 through GPIO3 | |
| 5:4 | **GPIO2 Route** — R/W. See bits 1:0 for description. |
| 3:2 | **GPIO1 Route** — R/W. See bits 1:0 for description. |
| 1:0 | **GPIO0 Route** — R/W. GPIO[15:0] can be routed to cause an SMI or SCI when the GPIO[n]_STS bit is set. If the GPIO0 is not set to an input, this field has no effect.<br>If the system is in an S1–S5 state and if the GPE0_EN bit is also set, then the GPIO can cause a Wake event, even if the GPIO is NOT routed to cause an SMI# or SCI.<br>00 = No effect.<br>01 = SMI# (if corresponding ALT_GPI_SMI_EN bit is also set)<br>10 = SCI (if corresponding GPE0_EN bit is also set)<br>11 = **Reserved** |

***Note:*** GPIOs that are not implemented will not have the corresponding bits implemented in this register.



## 13.8.2 APM I/O Decode

Table 13-128 shows the I/O registers associated with APM support. This register space is enabled in the PCI Device 31: Function 0 space (APMDEC_EN), and cannot be moved (fixed I/O location).

**Table 13-128.APM Register Map**

| Address | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| B2h | APM_CNT | Advanced Power Management Control Port | 00h | R/W |
| B3h | APM_STS | Advanced Power Management Status Port | 00h | R/W |

### 13.8.2.1 APM_CNT—Advanced Power Management Control Port Register

I/O Address: B2h
Default Value: 00h
Lockable: No
Power Well: Core
Attribute: R/W
Size: 8-bit
Usage: Legacy Only

| Bit | Description |
|---|---|
| 7:0 | Used to pass an APM command between the OS and the SMI handler. Writes to this port not only store data in the APMC register, but also generates an SMI# when the APMC_EN bit is set. |

### 13.8.2.2 APM_STS—Advanced Power Management Status Port Register

I/O Address: B3h
Default Value: 00h
Lockable: No
Power Well: Core
Attribute: R/W
Size: 8-bit
Usage: Legacy Only

| Bit | Description |
|---|---|
| 7:0 | Used to pass data between the OS and the SMI handler. Basically, this is a scratchpad register and is not affected by any other register or function (other than a PCI reset). |

## 13.8.3 Power Management I/O Registers

Table 13-129 shows the registers associated with ACPI and Legacy power management support. These registers are enabled in the PCI Device 31: Function 0 space (PM_IO_EN), and can be moved to any I/O location (128-byte aligned). The registers are defined to support the ACPI 2.0 specification, and use the same bit names.

***Note:*** All reserved bits and registers will always return 0 when read, and will have no effect when written.

### Table 13-129.ACPI and Legacy I/O Register Map (Sheet 1 of 2)

| PMBASE + Offset | Mnemonic | Register Name | ACPI Pointer | Default | Type |
|---|---|---|---|---|---|
| 00h–01h | PM1_STS | PM1 Status | PM1a_EVT_BLK | 0000h | R/WC |
| 02h–03h | PM1_EN | PM1 Enable | PM1a_EVT_BLK+2 | 0000h | R/W |
| 04h–07h | PM1_CNT | PM1 Control | PM1a_CNT_BLK | 00000000h | R/W, WO |
| 08h–0Bh | PM1_TMR | PM1 Timer | PMTMR_BLK | xx000000h | RO |
| 0Ch–0Fh | — | **Reserved** | — | — | — |
| 10h–13h | PROC_CNT | Processor Control | P_BLK | 00000000h | R/W, RO, WO |
| 14h–16h | — | **Reserved** (Nettop Only) | — | — | — |
| 14h | LV2 | Level 2 (Netbook Only) | P_BLK+4 | 00h | RO |
| 15h | LV3 | Level 3 (Netbook Only) | P_BLK+5 | 00h | RO |
| 16h | LV4 | Level 4 (Netbook Only) | P_BLK+6 | 00h | RO |
| 17h–1Fh | — | **Reserved** | — | — | — |
| 20h | — | **Reserved** (Nettop Only) | — | — | — |
| 20h | PM2_CNT | PM2 Control (Netbook Only) | PM2a_CNT_BLK | 00h | R/W |
| 28h–2Bh | GPE0_STS | General Purpose Event 0 Status | GPE0_BLK | 00000000h | R/WC |
| 2Ch–2Fh | GPE0_EN | General Purpose Event 0 Enables | GPE0_BLK+4 | 00000000h | R/W |
| 30h–33h | SMI_EN | SMI# Control and Enable | | 00000000h | R/W, WO, R/W (special) |
| 34h–37h | SMI_STS | SMI Status | | 00000000h | R/WC, RO |
| 38h–39h | ALT_GP_SMI_EN | Alternate GPI SMI Enable | | 0000h | R/W |
| 3Ah–3Bh | ALT_GP_SMI_STS | Alternate GPI SMI Status | | 0000h | R/WC |
| 3Ch–41h | — | **Reserved** | — | — | — |
| 42h | GPE_CNTL | General Purpose Event Control | | 00h | RO, R/W |
| 43h | — | **Reserved** | — | — | — |
| 44h–45h | DEVACT_STS | Device Activity Status | | 0000h | R/WC |
| 46h–4Fh | — | **Reserved** | | | |

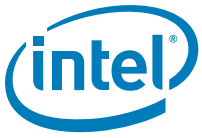


**Table 13-129.ACPI and Legacy I/O Register Map (Sheet 2 of 2)**

| PMBASE + Offset | Mnemonic | Register Name | ACPI Pointer | Default | Type |
|---|---|---|---|---|---|
| 50h | — | **Reserved** (Nettop Only) | | | |
| 50h | SS_CNT | Intel SpeedStep® Technology Control (Netbook Only) | | 01h | R/W (special) |
| 51h–5Fh | — | **Reserved** | — | — | — |
| 54h–57h | C3_RES | C3-Residency Register (Netbook Only) | — | 00000000h | RO, R/W |
| 60h–7Fh | — | **Reserved** for TCO | — | — | — |

### 13.8.3.1 PM1_STS—Power Management 1 Status Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 00h (*ACPI PM1a_EVT_BLK*) | Attribute: | R/WC |
| Default Value: | 0000h | Size: | 16-bit |
| Lockable: | No | Usage: | ACPI or Legacy |
| Power Well: | Bits 0–7: Core, Bits 8–15: Resume, except Bit 11 in RTC | | |

If bit 10 or 8 in this register is set, and the corresponding _EN bit is set in the PM1_EN register, then the Chipset will generate a Wake Event. Once back in an S0 state (or if already in an S0 state when the event occurs), the Chipset will also generate an SCI if the SCI_EN bit is set, or an SMI# if the SCI_EN bit is not set.

***Note:*** Bit 5 does not cause an SMI# or a wake event. Bit 0 does not cause a wake event but can cause an SMI# or SCI.

| Bit | Description |
|---|---|
| 15 | **Wake Status (WAK_STS)** — R/WC. This bit is not affected by hard resets caused by a CF9 write, but is reset by RSMRST#.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Set by hardware when the system is in one of the sleep states (via the SLP_EN bit) and an enabled wake event occurs. Upon setting this bit, the Chipset will transition the system to the ON state.<br>If the AFTERG3_EN bit is not set and a power failure (such as removed batteries on a Netbook platform) occurs without the SLP_EN bit set, the system will return to an S0 state when power returns, and the WAK_STS bit will not be set.<br>If the AFTERG3_EN bit is set and a power failure occurs without the SLP_EN bit having been set, the system will go into an S5 state when power returns, and a subsequent wake event will cause the WAK_STS bit to be set. Note that any subsequent wake event would have to be caused by either a Power Button press, or an enabled wake event that was preserved through the power failure (enable bit in the RTC well). |

| Bit | Description |
|---|---|
| 14 | PCI Express Wake Status (PCIEXPWAK_STS) — R/WC.<br>0 = Software clears this bit by writing a 1 to it. If the WAKE# pin is still active during the write or the PME message received indication has not been cleared in the root port, then the bit will remain active (i.e. all inputs to this bit are level-sensitive).<br>1 = This bit is set by hardware to indicate that the system woke due to a PCI Express wakeup event. This wakeup event can be caused by the PCI Express WAKE# pin being active or receipt of a PCI Express PME message at a root port. This bit is set only when one of these events causes the system to transition from a non-S0 system power state to the S0 system power state. This bit is set independent of the state of the PCIEXP_WAKE_DIS bit.<br><br>**NOTE:** This bit does not itself cause a wake event or prevent entry to a sleeping state. Thus, if the bit is 1 and the system is put into a sleeping state, the system will not automatically wake. |
| 14 | **Reserved** |
| 13:12 | **Reserved** |
| 11 | **Power Button Override Status (PRBTNOR_STS)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit is set any time a Power Button Override occurs (i.e., the power button is pressed for at least 4 consecutive seconds), or due to the corresponding bit in the SMBus slave message. The power button override causes an unconditional transition to the S5 state, as well as sets the AFTERG# bit. The BIOS or SCI handler clears this bit by writing a 1 to it. This bit is not affected by hard resets via CF9h writes, and is not reset by RSMRST#. Thus, this bit is preserved through power failures. Note that if this bit is still asserted when the global SCI_EN is set then an SCI will be generated. |
| 10 | **RTC Status (RTC_STS)** — R/WC. This bit is not affected by hard resets caused by a CF9 write, but is reset by RSMRST#.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Set by hardware when the RTC generates an alarm (assertion of the IRQ8# signal). Additionally if the RTC_EN bit (PMBASE + 02h, bit 10) is set, the setting of the RTC_STS bit will generate a wake event. |
| 9 | **Reserved** |

| Bit | Description |
|---|---|
| 8 | **Power Button Status (PWRBTN__STS)** — R/WC. This bit is not affected by hard resets caused by a CF9 write.<br>0 = If the PWRBTN# signal is held low for more than 4 seconds, the hardware clears the PWRBTN_STS bit, sets the PWRBTNOR_STS bit, and the system transitions to the S5 state with only PWRBTN# enabled as a wake event.<br><br>This bit can be cleared by software by writing a one to the bit position.<br>1 = This bit is set by hardware when the PWRBTN# signal is asserted Low, independent of any other enable bit.<br><br>In the S0 state, while PWRBTN_EN and PWRBTN_STS are both set, an SCI (or SMI# if SCI_EN is not set) will be generated.<br><br>In any sleeping state S1–S5, while PWRBTN_EN (PMBASE + 02h, bit 8) and PWRBTN_STS are both set, a wake event is generated.<br><br>**NOTE:** If the PWRBTN_STS bit is cleared by software while the PWRBTN# signal is sell asserted, this will not cause the PWRBN_STS bit to be set. The PWRBTN# signal must go inactive and active again to set the PWRBTN_STS bit. |
| 7:6 | **Reserved** |
| 5 | **Global Status (GBL _STS)** — R/WC.<br>0 = The SCI handler should then clear this bit by writing a 1 to the bit location.<br>1 = Set when an SCI is generated due to BIOS wanting the attention of the SCI handler. BIOS has a corresponding bit, BIOS_RLS, which will cause an SCI and set this bit. |
| 4 (Nettop Only) | **Reserved** |
| 4 (Netbook Only) | Bus Master Status (BM_STS) — R/WC. This bit will not cause a wake event, SCI or SMI#.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Set by the Chipset on Netbook platform when a bus master requests access to main memory. Bus master activity is detected by any of the PCI Requests being active, any internal bus master request being active, the BM_BUSY# signal being active, or REQ-C2 message received while in C3 or C4 state.<br><br>**NOTES:**<br>1. If the BM_STS_ZERO_EN bit is set, then this bit will generally report as a 0. LPC DMA (Netbook Only) and bus master activity will always set the BM_STS bit, even if the BM_STS_ZERO_EN bit is set. |
| 3:1 | **Reserved** |
| 0 | **Timer Overflow Status (TMROF_STS)** — R/WC.<br>0 = The SCI or SMI# handler clears this bit by writing a 1 to the bit location.<br>1 = This bit gets set any time bit 22 of the 24-bit timer goes high (bits are numbered from 0 to 23). This will occur every 2.3435 seconds. When the TMROF_EN bit (PMBASE + 02h, bit 0) is set, then the setting of the TMROF_STS bit will additionally generate an SCI or SMI# (depending on the SCI_EN). |

### 13.8.3.2 PM1_EN—Power Management 1 Enable Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 02h (*ACPI PM1a_EVT_BLK + 2*) | Attribute: | R/W |
| Default Value: | 0000h | Size: | 16-bit |
| Lockable: | No | Usage: | ACPI or Legacy |
| Power Well: | Bits 0–7: Core, Bits 8–9, 11–15: Resume, Bit 10: RTC | | |

| Bit | Description |
|---|---|
| 15 | **Reserved** |
| 14 | PCI Express Wake Disable(PCIEXPWAK_DIS) — R/W. Modification of this bit has no impact on the value of the PCIEXP_WAKE_STS bit.<br>0 = Inputs to the PCIEXP_WAKE_STS bit in the PM1 Status register enabled to wake the system.<br>1 = Inputs to the PCIEXP_WAKE_STS bit in the PM1 Status register disabled from waking the system. |
| 14 | **Reserved** |
| 13:11 | **Reserved** |
| 10 | **RTC Event Enable (RTC_EN)** — R/W. This bit is in the RTC well to allow an RTC event to wake after a power failure. This bit is not cleared by any reset other than RTCRST# or a Power Button Override event.<br>0 = No SCI (or SMI#) or wake event is generated then RTC_STS (PMBASE + 00h, bit 10) goes active.<br>1 = An SCI (or SMI#) or wake event will occur when this bit is set and the RTC_STS bit goes active. |
| 9 | **Reserved**. |
| 8 | **Power Button Enable (PWRBTN_EN)** — R/W. This bit is used to enable the setting of the PWRBTN_STS bit to generate a power management event (SMI#, SCI). PWRBTN_EN has no effect on the PWRBTN_STS bit (PMBASE + 00h, bit 8) being set by the assertion of the power button. The Power Button is always enabled as a Wake event.<br>0 = Disable.<br>1 = Enable. |
| 7:6 | **Reserved**. |
| 5 | **Global Enable (GBL_EN)** — R/W. When both the GBL_EN and the GBL_STS bit (PMBASE + 00h, bit 5) are set, an SCI is raised.<br>0 = Disable.<br>1 = Enable SCI on GBL_STS going active. |
| 4:1 | **Reserved**. |
| 0 | **Timer Overflow Interrupt Enable (TMROF_EN)** — R/W. Works in conjunction with the SCI_EN bit (PMBASE + 04h, bit 0) as described below: |

| TMROF_EN | SCI_EN | Effect when TMROF_STS is set |
|---|---|---|
| 0 | X | No SMI# or SCI |
| 1 | 0 | SMI# |
| 1 | 1 | SCI |

### 13.8.3.3 PM1_CNT—Power Management 1 Control

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 04h (*ACPI PM1a_CNT_BLK*) | Attribute: | R/W, WO |
| Default Value: | 00000000h | Size: | 32-bit |
| Lockable: | No | Usage: | ACPI or Legacy |
| Power Well: | Bits 0–7: Core, Bits 8–12: RTC, Bits 13–15: Resume | | |

| Bit | Description |
|---|---|
| 31:14 | **Reserved**. |
| 13 | **Sleep Enable** (**SLP_EN)** — WO. Setting this bit causes the system to sequence into the Sleep state defined by the SLP_TYP field. |
| 12:10 | **Sleep Type (SLP_TYP)** — R/W. This 3-bit field defines the type of Sleep the system should enter when the SLP_EN bit is set to 1. These bits are only reset by RTCRST#. |

| Code | Master Interrupt |
|---|---|
| 000b | ON: Typically maps to S0 state. |
| 001b | Asserts STPCLK#. Puts processor in Stop-Grant state. Optional to assert CPUSLP# to put processor in sleep state: Typically, maps to S1 state. |
| 010b | **Reserved** |
| 011b | **Reserved** |
| 100b | **Reserved** |
| 101b | Suspend-To-RAM. Assert SLP_S3#: Typically maps to S3 state. |
| 110b | Suspend-To-Disk. Assert SLP_S3#, and SLP_S4#: Typically maps to S4 state. |
| 111b | Soft Off. Assert SLP_S3#, SLP_S4#, and SLP_S5#: Typically maps to S5 state. |

| | |
|---|---|
| 9:3 | **Reserved**. |
| 2 | **Global Release (GBL_RLS)** — WO.<br>0 = This bit always reads as 0.<br>1 = ACPI software writes a 1 to this bit to raise an event to the BIOS. BIOS software has a corresponding enable and status bits to control its ability to receive ACPI events. |

| Bit | Description |
|---|---|
| 1 (Nettop Only) | **Reserved** |
| 1 (Netbook Only) | **Bus Master Reload (BM_RLD)** — R/W. This bit is treated as a scratchpad bit. This bit is reset to 0 by PLTRST#<br>0 = Bus master requests will not cause a break from the C3 state.<br>1 = Enable Bus Master requests (internal, external or BM_BUSY#) to cause a break from the C3 state.<br>If software fails to set this bit before going to C3 state, the Chipset on Netbook platform will still return to a snoopable state from C3 or C4 states due to bus master activity. |
| 0 | **SCI Enable** (**SCI_EN)** — R/W. Selects the SCI interrupt or the SMI# interrupt for various events including the bits in the PM1_STS register (bit 10, 8, 0), and bits in GPE0_STS.<br>0 = These events will generate an SMI#.<br>1 = These events will generate an SCI. |

### 13.8.3.4 PM1_TMR—Power Management 1 Timer Register

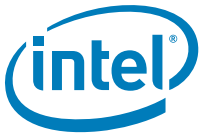
I/O Address: PMBASE + 08h (*ACPI PMTMR_BLK*)
Attribute: RO
Default Value: xx000000h
Size: 32-bit
Lockable: No
Usage: ACPI
Power Well: Core

| Bit | Description |
|---|---|
| 31:24 | **Reserved** |
| 23:0 | **Timer Value (TMR_VAL)** — RO. Returns the running count of the PM timer. This counter runs off a 3.579545 MHz clock (14.31818 MHz divided by 4). It is reset to 0 during a PCI reset, and then continues counting as long as the system is in the S0 state. After an S1 state, the counter will not be reset (it will continue counting from the last value in S0 state.<br>Anytime bit 22 of the timer goes HIGH to LOW (bits referenced from 0 to 23), the TMROF_STS bit (PMBASE + 00h, bit 0) is set. The High-to-Low transition will occur every 2.3435 seconds. If the TMROF_EN bit (PMBASE + 02h, bit 0) is set, an SCI interrupt is also generated. |

### 13.8.3.5 PROC_CNT—Processor Control Register

I/O Address: PMBASE + 10h (*ACPI P_BLK*)
Attribute: R/W, RO, WO
Default Value: 00000000h
Size: 32-bit
Lockable: No (bits 7:5 are write once)
Usage: ACPI or Legacy
Power Well: Core

| Bit | Description |
|---|---|
| 31:18 | **Reserved** |



<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>17</td><td>Throttle Status (THTL_STS) — RO.<br>0 = No clock throttling is occurring (maximum processor performance).<br>1 = Indicates that the clock state machine is throttling the processor performance. This could be due to the THT_EN bit or the FORCE_THTL bit being set.</td></tr>
<tr><td>16:9</td><td>Reserved</td></tr>
<tr><td>8</td><td>Force Thermal Throttling (FORCE_THTL) — R/W. Software can set this bit to force the thermal throttling function.<br>0 = No forced throttling.<br>1 = Throttling at the duty cycle specified in THRM_DTY starts immediately, and no SMI# is generated.</td></tr>
<tr><td>7:5</td><td>THRM_DTY — WO. This write-once field determines the duty cycle of the throttling when the FORCE_THTL bit is set. The duty cycle indicates the approximate percentage of time the STPCLK# signal is asserted while in the throttle mode. The STPCLK# throttle period is 1024 PCICLKs. Note that the throttling only occurs if the system is in the C0 state. For Netbook only, If in the C2, C3, or C4 state, no throttling occurs.<br>Once the THRM_DTY field is written, any subsequent writes will have no effect until PLTRST# goes active.
<table>
<tr><th>THRM_DTY</th><th>Throttle Mode</th><th>PCI Clocks</th></tr>
<tr><td>000b</td><td>50% (Default)</td><td>512</td></tr>
<tr><td>001b</td><td>87.5%</td><td>896</td></tr>
<tr><td>010b</td><td>75.0%</td><td>768</td></tr>
<tr><td>011b</td><td>62.5%</td><td>640</td></tr>
<tr><td>100b</td><td>50%</td><td>512</td></tr>
<tr><td>101b</td><td>37.5%</td><td>384</td></tr>
<tr><td>110b</td><td>25%</td><td>256</td></tr>
<tr><td>111b</td><td>12.5%</td><td>128</td></tr>
</table></td></tr>
<tr><td>4</td><td>THTL_EN — R/W. When set and the system is in a C0 state, it enables a processor-controlled STPCLK# throttling. The duty cycle is selected in the THTL_DTY field.<br>0 = Disable<br>1 = Enable</td></tr>
<tr><td>3:1</td><td>THTL_DTY — R/W. This field determines the duty cycle of the throttling when the THTL_EN bit is set. The duty cycle indicates the approximate percentage of time the STPCLK# signal is asserted (low) while in the throttle mode. The STPCLK# throttle period is 1024 PCICLKs.
<table>
<tr><th>THTL_DTY</th><th>Throttle Mode</th><th>PCI Clocks</th></tr>
<tr><td>000b</td><td>50% (Default)</td><td>512</td></tr>
<tr><td>001b</td><td>87.5%</td><td>896</td></tr>
<tr><td>010b</td><td>75.0%</td><td>768</td></tr>
<tr><td>011b</td><td>62.5%</td><td>640</td></tr>
<tr><td>100b</td><td>50%</td><td>512</td></tr>
<tr><td>101b</td><td>37.5%</td><td>384</td></tr>
<tr><td>110b</td><td>25%</td><td>256</td></tr>
<tr><td>111b</td><td>12.5%</td><td>128</td></tr>
</table></td></tr>
<tr><td>0</td><td>Reserved</td></tr>
</table>

### 13.8.3.6 LV2 — Level 2 Register (Netbook Only)

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 14h (*ACPI P_BLK+4*) | Attribute: | RO |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Usage: | ACPI or Legacy |
| Power Well: | Core | | |

| Bit | Description |
|---|---|
| 7:0 | Reads to this register return all 0s, writes to this register have no effect. Reads to this register generate a “enter a level 2 power state” (C2) to the clock control logic. This will cause the STPCLK# signal to go active, and stay active until a break event occurs. Throttling (due either to THTL_EN or FORCE_THTL) will be ignored. |

**NOTE:** This register should not be used by Intel® iA64 processors or systems with more than 1 logical processor, unless appropriate semaphoring software has been put in place to ensure that all threads/processors are ready for the C2 state when the “read to this register” instruction occurs.

### 13.8.3.7 LV3—Level 3 Register (Netbook Only)

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 15h (*ACPI P_BLK + 5*) | Attribute: | RO |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Usage: | ACPI or Legacy |
| | | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | Reads to this register return all 0s, writes to this register have no effect. Reads to this register generate a “enter a C3 power state” to the clock control logic. The C3 state persists until a break event occurs. |

**NOTES:**
1. If the C4onC3_EN bit is set, reads this register will initiate a LVL4 transition rather than a LVL3 transition. In the event that software attempts to simultaneously read the LVL2 and LVL3 registers (which is not permitted), the Chipset on Netbook platform will ignore the LVL3 read, and only perform a C2 transition.
2. This register should not be used by iA64 processors or systems with more than 1 logical processor, unless appropriate semaphoring software has been put in place to ensure that all threads/processors are ready for the C3 state when the “read to this register” instruction occurs.

### 13.8.3.8 LV4—Level 4 Register (Netbook Only)

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 16h (*ACPI P_BLK + 6*) | Attribute: | RO |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Usage: | ACPI or Legacy |
| | | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | Reads to this register return all 0s, writes to this register have no effect. Reads to this register generate a “enter a C4 power state” to the clock control logic. The C4 state persists until a break event occurs. |

**NOTE:** This register should not be used by iA64 processors or systems with more than 1 logical processor, unless appropriate semaphoring software has been put in place to ensure that all threads/processors are ready for the C4 state when the “read to this register” instruction occurs.

### 13.8.3.9 PM2_CNT—Power Management 2 Control Register (Netbook Only)

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 20h (*ACPI PM2_BLK*) | Attribute: | R/W |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Usage: | ACPI |
| Power Well: | Core | | |

| Bit | Description |
|---|---|
| 7:1 | **Reserved** |
| 0 | **Arbiter Disable (ARB_DIS)** — R/W. This bit is a scratchpad bit for legacy software compatibility. Software typically sets this bit to 1 prior to entering a C3 or C4 state. When a transition to a C3 or C4 state occurs, Chipset on Netbook platform will automatically prevent any internal or external non-Isoch bus masters from initiating any cycles up to the (G)MCH/CPU. This blocking starts immediately upon the Chipset sending the Go–C3 message to the (G)MCH/CPU. The blocking stops when the Ack-C2 message is received. Note that this is not really blocking, in that messages (such as from PCI Express*) are just queued and held pending. |

### 13.8.3.10 GPE0_STS—General Purpose Event 0 Status Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 28h (*ACPI GPE0_BLK*) | Attribute: | R/WC |
| Default Value: | 00000000h | Size: | 32-bit |
| Lockable: | No | Usage: | ACPI |
| Power Well: | Resume | | |

This register is symmetrical to the General Purpose Event 0 Enable Register. Unless indicated otherwise below, if the corresponding _EN bit is set, then when the _STS bit get set, the Chipset will generate a Wake Event. Once back in an S0 state (or if already in an S0 state when the event occurs), the Chipset will also generate an SCI if the SCI_EN bit is set, or an SMI# if the SCI_EN bit (PMBASE + 04h, bit 0) is not set. Bits 31:16 are reset by a CF9h write; bits 15:0 are not. All are reset by RSMRST#.

| Bit | Description |
|---|---|
| 31:16 | **GPIOn_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = These bits are set any time the corresponding GPIO is set up as an input and the corresponding GPIO signal is high (or low if the corresponding GP_INV bit is set). If the corresponding enable bit is set in the GPE0_EN register, then when the GPIO[n]_STS bit is set:<br>• If the system is in an S1–S5 state, the event will also wake the system.<br>• If the system is in an S0 state (or upon waking back to an S0 state), a SCI will be caused depending on the GPIO_ROUT bits (D31:F0:B8h, bits 31:30) for the corresponding GPI.<br><br>**NOTE:** Mapping is as follows: bit 31 corresponds to GPIO15... and bit 16 corresponds to GPIO0. |
| 15 | **Reserved** |
| 14 | **USB4_STS** — R/WC.<br>0 = Disable.<br>1 = Set by hardware and can be reset by writing a one to this bit position or a resume well reset. This bit is set when USB UHCI controller #4 needs to cause a wake. Additionally if the USB4_EN bit is set, the setting of the USB4_STS bit will generate a wake event. |
| 13 | **PME_B0_STS** — R/WC. This bit will be set to 1 by the Chipset when any internal device with PCI Power Management capabilities on bus 0 asserts the equivalent of the PME# signal. Additionally, if the PME_B0_EN bit is set, and the system is in an S0 state, then the setting of the PME_B0_STS bit will generate an SCI (or SMI# if SCI_EN is not set). If the PME_B0_STS bit is set, and the system is in an S1–S4 state (or S5 state due to SLP_TYP and SLP_EN), then the setting of the PME_B0_STS bit will generate a wake event, and an SCI (or SMI# if SCI_EN is not set) will be generated. If the system is in an S5 state due to power button override, then the PME_B0_STS bit will not cause a wake event or SCI.<br><br>The default for this bit is 0. Writing a 1 to this bit position clears this bit. |
| 12 | **USB3_STS** — R/WC.<br>0 = Disable.<br>1 = Set by hardware and can be reset by writing a one to this bit position or a resume well reset. This bit is set when USB UHCI controller #3 needs to cause a wake. Additionally if the USB3_EN bit is set, the setting of the USB3_STS bit will generate a wake event. |
| 11 | **PME_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Set by hardware when the PME# signal goes active. Additionally, if the PME_EN bit is set, and the system is in an S0 state, then the setting of the PME_STS bit will generate an SCI or SMI# (if SCI_EN is not set). If the PME_EN bit is set, and the system is in an S1–S4 state (or S5 state due to setting SLP_TYP and SLP_EN), then the setting of the PME_STS bit will generate a wake event, and an SCI will be generated. If the system is in an S5 state due to power button override or a power failure, then PME_STS will not cause a wake event or SCI. |
| 10 (Nettop Only) | **Reserved** |

| Bit | Description |
|---|---|
| 10 (Netbook Only) | **BATLOW_STS** — R/WC. (Netbook Only) Software clears this bit by writing a 1 to it.<br>0 = BATLOW# Not asserted<br>1 = Set by hardware when the BATLOW# signal is asserted. |
| 9 | **PCI_EXP_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Set by hardware to indicate that:<br>• The PME event message was received on one or more of the PCI Express* ports<br>• An Assert PMEGPE message received from the (G)MCH/CPU via DMI<br><br>**NOTES:**<br>1. The PCI WAKE# pin has no impact on this bit.<br>2. If the PCI_EXP_STS bit went active due to an Assert PMEGPE message, then a Deassert PMEGPE message must be received prior to the software write in order for the bit to be cleared.<br>3. If the bit is not cleared and the corresponding PCI_EXP_EN bit is set, the level-triggered SCI will remain active.<br>4. A race condition exists where the PCI Express device sends another PME message because the PCI Express device was not serviced within the time when it must resend the message. This may result in a spurious interrupt, and this is comprehended and approved by the *PCI Express* Specification, Revision 1.0a*. The window for this race condition is approximately 95–105 milliseconds. |
| 8 | **RI_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Set by hardware when the RI# input signal goes active. |
| 7 | **SMBus Wake Status (SMB_WAK_STS)** — R/WC. The SMBus controller can independently cause an SMI# or SCI, so this bit does not need to do so (unlike the other bits in this register). Software clears this bit by writing a 1 to it.<br>0 = Wake event Not caused by the chipset's SMBus logic.<br>1 = Set by hardware to indicate that the wake event was caused by the chipset's SMBus logic.This bit will be set by the WAKE/SMI# command type, even if the system is already awake. The SMI handler should then clear this bit.<br><br>**NOTES:**<br>1. This bit is set by the SMBus slave command 01h (Wake/SMI#) even when the system is in the S0 state. Therefore, to avoid an instant wake on subsequent transitions to sleep states, software must clear this bit after each reception of the Wake/SMI# command or just prior to entering the sleep state.<br>2. If SMB_WAK_STS is set due to SMBus slave receiving a message, it will be cleared by internal logic when a THRMTRIP# event happens or a Power Button Override event. However, THRMTRIP# or Power Button Override event will not clear SMB_WAK_STS if it is set due to SMBALERT# signal going active.<br>3. The SMBALERT_STS bit (D31:F3:I/O Offset 00h:Bit 5) should be cleared by software before the SMB_WAK_STS bit is cleared. |
| 6 | **TCOSCI_STS** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = TOC logic did Not cause SCI.<br>1 = Set by hardware when the TCO logic causes an SCI. |

| Bit | Description |
|---|---|
| 5 | **AC97_STS** — R/WC. This bit will be set to 1 when the codecs are attempting to wake the system and the PME events for the codecs are armed for wakeup. A PME is armed by programming the appropriate PMEE bit in the Power Management Control and Status register at bit 8 of offset 54h in each AC'97 function.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Set by hardware when the codecs are attempting to wake the system. The AC97_STS bit gets set only from the following case:<br>1.The PMEE bit for the function is set, and The AC-link bit clock has been shut and the routed ACZ_SDIN line is high (for audio, if routing is disabled, no wake events are allowed.<br>NOTE: This bit is not affected by a hard reset caused by a CF9h write. |
| 4 | **USB2_STS** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = USB UHCI controller 2 does Not need to cause a wake.<br>1 = Set by hardware when USB UHCI controller 2 needs to cause a wake. Wake event will be generated if the corresponding USB2_EN bit is set. |
| 3 | **USB1_STS** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = USB UHCI controller 1 does Not need to cause a wake.<br>1 = Set by hardware when USB UHCI controller 1 needs to cause a wake. Wake event will be generated if the corresponding USB1_EN bit is set. |
| 2 | SWGPE_STS — R/WC.<br>The SWGPE_CTRL bit (bit 1 of GPE_CTRL reg) acts as a level input to this bit. |
| 1 | **HOT_PLUG_STS** — R/WC.<br>0 = This bit is cleared by writing a 1 to this bit position.<br>1 = When a PCI Express* Hot-Plug event occurs. This will cause an SCI if the HOT_PLUG_EN bit is set in the GEP0_EN register. |
| 1 | **Reserved** |
| 0 | **Thermal Interrupt Status (THRM_STS)** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = THRM# signal Not driven active as defined by the THRM_POL bit<br>1 = Set by hardware anytime the THRM# signal is driven active as defined by the THRM_POL bit. Additionally, if the THRM_EN bit is set, then the setting of the THRM_STS bit will also generate a power management event (SCI or SMI#). |

### 13.8.3.11 GPE0_EN—General Purpose Event 0 Enables Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 2Ch (*ACPI GPE0_BLK + 4*) | Attribute: | R/W |
| Default Value: | 00000000h | Size: | 32-bit |
| Lockable: | No | Usage: | ACPI |
| Power Well: | Bits 0–7, 9, 12, 14–31 Resume, Bits 8, 10–11, 13 RTC | | |

This register is symmetrical to the General Purpose Event 0 Status Register. All the bits in this register should be cleared to 0 based on a Power Button Override or processor Thermal Trip event. The resume well bits are all cleared by RSMRST#. The RTC sell bits are cleared by RTCRST#.

| Bit | Description |
|---|---|
| 31:16 | **GPIn_EN** — R/W. These bits enable the corresponding GPI[n]_STS bits being set to cause a SCI, and/or wake event. These bits are cleared by RSMRST#.<br>**NOTE:** Mapping is as follows: bit 31 corresponds to GPIO15... and bit 16 corresponds to GPIO0. |
| 15 | **Reserved** |
| 14 | **USB4_EN** — R/W.<br>0 = Disable.<br>1 = Enable the setting of the USB4_STS bit to generate a wake event. The USB4_STS bit is set anytime USB UHCI controller #4 signals a wake event. Break events are handled via the USB interrupt. |
| 13 | **PME_B0_EN** — R/W.<br>0 = Disable<br>1 = Enables the setting of the PME_B0_STS bit to generate a wake event and/or an SCI or SMI#. PME_B0_STS can be a wake event from the S1–S4 states, or from S5 (if entered via SLP_TYP and SLP_EN) or power failure, but not Power Button Override. This bit defaults to 0.<br><br>**NOTE:** It is only cleared by Software or RTCRST#. It is not cleared by CF9h writes. |
| 12 | **USB3_EN** — R/W.<br>0 = Disable.<br>1 = Enable the setting of the USB3_STS bit to generate a wake event. The USB3_STS bit is set anytime USB UHCI controller #3 signals a wake event. Break events are handled via the USB interrupt. |
| 11 | **PME_EN** — R/W.<br>0 = Disable.<br>1 = Enables the setting of the PME_STS to generate a wake event and/or an SCI. PME# can be a wake event from the S1 – S4 state or from S5 (if entered via SLP_EN, but not power button override). |
| 10 (Nettop Only) | **Reserved** |
| 10 (Netbook Only) | **BATLOW_EN** — R/W. (Netbook Only)<br>0 = Disable.<br>1 = Enables the BATLOW# signal to cause an SMI# or SCI (depending on the SCI_EN bit) when it goes low. This bit does not prevent the BATLOW# signal from inhibiting the wake event. |
| 9 | **PCI_EXP_EN** — R/W.<br>0 = Disable SCI generation upon PCI_EXP_STS bit being set.<br>1 = Enables Chipset to cause an SCI when PCI_EXP_STS bit is set. This is used to allow the PCI Express* ports, including the link to the (G)MCH/CPU, to cause an SCI due to wake/PME events. |
| 9 | **Reserved**. Must be programmed to 0. |
| 8 | **RI_EN** — R/W. The value of this bit will be maintained through a G3 state and is not affected by a hard reset caused by a CF9h write.<br>0 = Disable.<br>1 = Enables the setting of the RI_STS to generate a wake event. |
| 7 | **Reserved** |

| Bit | Description |
|---|---|
| 6 | **TCOSCI_EN** — R/W.<br>0 = Disable.<br>1 = Enables the setting of the TCOSCI_STS to generate an SCI. |
| 5 | AC97_EN — R/W.<br>0 = Disable.<br>1 = Enables the setting of the AC97_STS to generate a wake event.<br>**NOTE:** This bit is also used for Intel HD Audio when the Intel High Definition Audio host controller is enabled. |
| 4 | **USB2_EN** — R/W.<br>0 = Disable.<br>1 = Enables the setting of the USB2_STS to generate a wake event. |
| 3 | **USB1_EN** — R/W.<br>0 = Disable.<br>1 = Enables the setting of the USB1_STS to generate a wake event. |
| 2 | SWGPE_EN— R/W. This bit allows software to control the assertion of SWGPE_STS bit. This bit This bit, when set to 1, enables the SW GPE function. If SWGPE_CTRL is written to a 1, hardware will set SWGPE_STS (acts as a level input)<br>If SWGPE_STS, SWGPE_EN, and SCI_EN are all 1s, an SCI will be generated<br>If SWGPE_STS = 1, SWGPE_EN = 1, SCI_EN = 0, and GBL_SMI_EN = 1, then an SMI# will be generated |
| 1 | **HOT_PLUG_EN** — R/W.<br>0 = Disables SCI generation upon the HOT_PLUG_STS bit being set.<br>1 = Enables the Chipset to cause an SCI when the HOT_PLUG_STS bit is set. This is used to allow the PCI Express ports to cause an SCI due to hot-plug events. |
| 1 | **Reserved** |
| 0 | **THRM_EN** — R/W.<br>0 = Disable.<br>1 = Active assertion of the THRM# signal (as defined by the THRM_POL bit) will set the THRM_STS bit and generate a power management event (SCI or SMI). |

### 13.8.3.12 SMI_EN—SMI Control and Enable Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 30h | Attribute: | R/W, R/W (special), WO |
| Default Value: | 00000000h | Size: | 32 bit |
| Lockable: | No | Usage: | ACPI or Legacy |
| Power Well: | Core | | |

***Note:*** This register is symmetrical to the SMI status register.

| Bit | Description |
|---|---|
| 31:26 | **Reserved** |
| 25 | **Reserved** |
| 24:19 | **Reserved** |
| 18 | **INTEL_USB2_EN** — R/W.<br>0 = Disable<br>1 = Enables Intel-Specific USB2 SMI logic to cause SMI#. |

| Bit | Description |
|---|---|
| 17 | **LEGACY_USB2_EN** — R/W.<br>0 = Disable<br>1 = Enables legacy USB2 logic to cause SMI#. |
| 16:15 | **Reserved** |
| 14 | **PERIODIC_EN** — R/W.<br>0 = Disable.<br>1 = Enables the Chipset to generate an SMI# when the PERIODIC_STS bit (PMBASE + 34h, bit 14) is set in the SMI_STS register (PMBASE + 34h). |
| 13 | **TCO_EN** — R/W.<br>0 = Disables TCO logic generating an SMI#. Note that if the NMI2SMI_EN bit is set, SMIs that are caused by re-routed NMIs will not be gated by the TCO_EN bit. Even if the TCO_EN bit is 0, NMIs will still be routed to cause SMIs.<br>1 = Enables the TCO logic to generate SMI#.<br><br>**NOTE:** This bit cannot be written once the TCO_LOCK bit is set. |
| 12 | **Reserved** |
| 11 | **MCSMI_ENMicrocontroller SMI Enable (MCSMI_EN)** — R/W.<br>0 = Disable.<br>1 = Enables Chipset to trap accesses to the microcontroller range (62h or 66h) and generate an SMI#. Note that “trapped’ cycles will be claimed by the Chipset on PCI, but not forwarded to LPC. |
| 10:8 | **Reserved** |
| 7 | **BIOS Release (BIOS_RLS)** — WO.<br>0 = This bit will always return 0 on reads. Writes of 0 to this bit have no effect.<br>1 = Enables the generation of an SCI interrupt for ACPI software when a one is written to this bit position by BIOS software.<br><br>**NOTE:** 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. |
| 6 | **Software SMI# Timer Enable (SWSMI_TMR_EN)** — R/W.<br>0 = Disable. Clearing the SWSMI_TMR_EN bit before the timer expires will reset the timer and the SMI# will not be generated.<br>1 = Starts Software SMI# Timer. When the SWSMI timer expires (the timeout period depends upon the SWSMI_RATE_SEL bit setting), SWSMI_TMR_STS is set and an SMI# is generated. SWSMI_TMR_EN stays set until cleared by software. |
| 5 | **APMC_EN** — R/W.<br>0 = Disable. Writes to the APM_CNT register will not cause an SMI#.<br>1 = Enables writes to the APM_CNT register to cause an SMI#. |
| 4 | **SLP_SMI_EN** — R/W.<br>0 = Disables the generation of SMI# on SLP_EN. Note that this bit must be 0 before the software attempts to transition the system into a sleep state by writing a 1 to the SLP_EN bit.<br>1 = A write of 1 to the SLP_EN bit (bit 13 in PM1_CNT register) will generate an SMI#, and the system will not transition to the sleep state based on that write to the SLP_EN bit. |

| Bit | Description |
|---|---|
| 3 | **LEGACY_USB_EN** — R/W.<br>0 = Disable.<br>1 = Enables legacy USB circuit to cause SMI#. |
| 2 | **BIOS_EN** — R/W.<br>0 = Disable.<br>1 = Enables the generation of SMI# when ACPI software writes a 1 to the GBL_RLS bit (D31:F0:PMBase + 04h:bit 2). Note that if the BIOS_STS bit (D31:F0:PMBase + 34h:bit 2), which gets set when software writes 1 to GBL_RLS bit, is already a 1 at the time that BIOS_EN becomes 1, an SMI# will be generated when BIOS_EN gets set. |
| 1 | **End of SMI (EOS)** — R/W (special). This bit controls the arbitration of the SMI signal to the processor. This bit must be set for the Chipset to assert SMI# low to the processor after SMI# has been asserted previously.<br>0 = Once the Chipset asserts SMI# low, the EOS bit is automatically cleared.<br>1 = When this bit is set to 1, SMI# signal will be deasserted for 4 PCI clocks before its assertion. In the SMI handler, the processor should clear all pending SMIs (by servicing them and then clearing their respective status bits), set the EOS bit, and exit SMM. This will allow the SMI arbiter to re-assert SMI upon detection of an SMI event and the setting of a SMI status bit.<br><br>**NOTE:** Chipset is able to generate 1st SMI after reset even though EOS bit is not set. Subsequent SMI require EOS bit is set. |
| 0 | **GBL_SMI_EN** — R/W.<br>0 = No SMI# will be generated by Chipset. This bit is reset by a PCI reset event.<br>1 = Enables the generation of SMI# in the system upon any enabled SMI event.<br>**NOTE:** When the SMI_LOCK bit is set, this bit cannot be changed. |

### 13.8.3.13 SMI_STS—SMI Status Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 34h | Attribute: | RO, R/WC |
| Default Value: | 00000000h | Size: | 32-bit |
| Lockable: | No | Usage: | ACPI or Legacy |
| Power Well: | Core | | |

***Note:*** If the corresponding _EN bit is set when the _STS bit is set, the Chipset will cause an SMI# (except bits 8–10 and 12, which do not need enable bits since they are logic ORs of other registers that have enable bits). The Chipset uses the same GPE0_EN register (I/O address: PMBase+2Ch) to enable/disable both SMI and ACPI SCI general purpose input events. ACPI OS assumes that it owns the entire GPE0_EN register per ACPI spec. Problems arise when some of the general-purpose inputs are enabled as SMI by BIOS, and some of the general purpose inputs are enabled for SCI. In this case ACPI OS turns off the enabled bit for any GPIx input signals that are not indicated as SCI general-purpose events at boot, and exit from sleeping states. BIOS should define a dummy control method which prevents the ACPI OS from clearing the SMI GPE0_EN bits.

| Bit | Description |
|---|---|
| 31:27 | **Reserved** |

| Bit | Description |
|---|---|
| 26 | SPI_STS — RO. This bit will be set if the SPI logic is generating an SMI#. This bit is read only because the sticky status and enable bits associated with this function are located in the SPI registers. |
| 25 | **Reserved** |
| 24:22 | **Reserved** |
| 21 | MONITOR_STS — RO. This bit will be set if the Trap/SMI logic has caused the SMI. This will occur when the processor or a bus master accesses an assigned register (or a sequence of accesses). See Section 10.1.32 through Section 10.1.35 for details on the specific cause of the SMI. |
| 20 | **PCI_EXP_SMI_STS** — RO. PCI Express* SMI event occurred. This could be due to a PCI Express PME event or Hot-Plug event. |
| 20:19 | **Reserved** |
| 19 | **Reserved** |
| 18 | **INTEL_USB2_STS** — RO. This non-sticky read-only bit is a logical OR of each of the SMI status bits in the Intel-Specific USB2 SMI Status Register ANDed with the corresponding enable bits. This bit will not be active if the enable bits are not set. Writes to this bit will have no effect. |
| 17 | **LEGACY_USB2_STS** — RO. This non-sticky read-only bit is a logical OR of each of the SMI status bits in the USB2 Legacy Support Register ANDed with the corresponding enable bits. This bit will not be active if the enable bits are not set. Writes to this bit will have no effect. |
| 16 | **SMBus SMI Status (SMBUS_SMI_STS)** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = This bit is set from the 64 kHz clock domain used by the SMBus. Software must wait at least 15.63 us after the initial assertion of this bit before clearing it.<br>1 = Indicates that the SMI# was caused by:<br>1. The SMBus Slave receiving a message that an SMI# should be caused, or<br>2. The SMBALERT# signal goes active and the SMB_SMI_EN bit is set and the SMBALERT_DIS bit is cleared, or<br>3. The SMBus Slave receiving a Host Notify message and the HOST_NOTIFY_INTREN and the SMB_SMI_EN bits are set, or<br>4. The Chipset detecting the SMLINK_SLAVE_SMI command while in the S0 state. |
| 15 | **SERIRQ_SMI_STS** — RO.<br>0 = SMI# was not caused by the SERIRQ decoder.<br>1 = Indicates that the SMI# was caused by the SERIRQ decoder.<br>**NOTE:** This is not a sticky bit |
| 14 | **PERIODIC_STS** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit is set at the rate determined by the PER_SMI_SEL bits. If the PERIODIC_EN bit (PMBASE + 30h, bit 14) is also set, the Chipset generates an SMI#. |
| 13 | TCO_STS — R/WC. Software clears this bit by writing a 1 to it.<br>0 = SMI# not caused by TCO logic.<br>1 = Indicates the SMI# was caused by the TCO logic. Note that this is not a wake event. |

| Bit | Description |
|---|---|
| 12 | **Device Monitor Status (DEVMON_STS)** — RO.<br>0 = SMI# not caused by Device Monitor.<br>1 = Set if bit 0 of the DEVACT_STS register (PMBASE + 44h) is set. The bit is not sticky, so writes to this bit will have no effect. |
| 11 | **Microcontroller SMI# Status** (**MCSMI_STS)** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = Indicates that there has been no access to the power management microcontroller range (62h or 66h).<br>1 = Set if there has been an access to the power management microcontroller range (62h or 66h) and the Microcontroller Decode Enable #1 bit in the LPC Bridge I/O Enables configuration register is 1 (D31:F0:Offset 82h:bit 11). Note that this implementation assumes that the Microcontroller is on LPC. If this bit is set, and the MCSMI_EN bit is also set, the Chipset will generate an SMI#. |
| 10 | **GPI_STS** — RO. This bit is a logical OR of the bits in the ALT_GP_SMI_STS register that are also set up to cause an SMI# (as indicated by the GPI_ROUT registers) and have the corresponding bit set in the ALT_GP_SMI_EN register. Bits that are not routed to cause an SMI# will have no effect on this bit.<br>0 = SMI# was not generated by a GPI assertion.<br>1 = SMI# was generated by a GPI assertion. |
| 9 | **GPE0_STS** — RO. This bit is a logical OR of the bits 14:10, 8:2, and 0 in the GPE0_STS register (PMBASE + 28h) that also have the corresponding bit set in the GPE0_EN register (PMBASE + 2Ch).<br>0 = SMI# was not generated by a GPE0 event.<br>1 = SMI# was generated by a GPE0 event. |
| 8 | **PM1_STS_REG** — RO. This is an ORs of the bits in the ACPI PM1 Status Register (offset PMBASE+00h) that can cause an SMI#.<br>0 = SMI# was not generated by a PM1_STS event.<br>1 = SMI# was generated by a PM1_STS event. |
| 7 | **Reserved** |
| 6 | **SWSMI_TMR_STS** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = Software SMI# Timer has Not expired.<br>1 = Set by the hardware when the Software SMI# Timer expires. |
| 5 | **APM_STS** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = No SMI# generated by write access to APM Control register with APMCH_EN bit set.<br>1 = SMI# was generated by a write access to the APM Control register with the APMC_EN bit set. |
| 4 | **SLP_SMI_STS** — R/WC. Software clears this bit by writing a 1 to the bit location.<br>0 = No SMI# caused by write of 1 to SLP_EN bit when SLP_SMI_EN bit is also set.<br>1 = Indicates an SMI# was caused by a write of 1 to SLP_EN bit when SLP_SMI_EN bit is also set. |
| 3 | **LEGACY_USB_STS** — RO. This bit is a logical OR of each of the SMI status bits in the USB Legacy Keyboard/Mouse Control Registers ANDed with the corresponding enable bits. This bit will not be active if the enable bits are not set.<br>0 = SMI# was not generated by USB Legacy event.<br>1 = SMI# was generated by USB Legacy event. |

| Bit | Description |
|---|---|
| 2 | **BIOS_STS** — R/WC.<br>0 = No SMI# generated due to ACPI software requesting attention.<br>1 = This bit gets set by hardware when a 1 is written by software to the GBL_RLS bit (D31:F0:PMBase + 04h:bit 2). When both the BIOS_EN bit (D31:F0:PMBase + 30h:bit 2) and the BIOS_STS bit are set, an SMI# will be generated. The BIOS_STS bit is cleared when software writes a 1 to its bit position. |
| 1:0 | **Reserved** |

### 13.8.3.14 ALT_GP_SMI_EN—Alternate GPI SMI Enable Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE +38h | Attribute: | R/W |
| Default Value: | 0000h | Size: | 16-bit |
| Lockable: | No | Usage: | ACPI or Legacy |
| Power Well: | Resume | | |

| Bit | Description |
|---|---|
| 15:0 | **Alternate GPI SMI Enable** — R/W. These bits are used to enable the corresponding GPIO to cause an SMI#. For these bits to have any effect, the following must be true.<br>• The corresponding bit in the ALT_GP_SMI_EN register is set.<br>• The corresponding GPI must be routed in the GPI_ROUT register to cause an SMI.<br>• The corresponding GPIO must be implemented.<br><br>**NOTE:** Mapping is as follows: bit 15 corresponds to GPIO15... bit 0 corresponds to GPIO0. |

### 13.8.3.15 ALT_GP_SMI_STS—Alternate GPI SMI Status Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE +3Ah | Attribute: | R/WC |
| Default Value: | 0000h | Size: | 16-bit |
| Lockable: | No | Usage: | ACPI or Legacy |
| Power Well: | Resume | | |

| Bit | Description |
|---|---|
| 15:0 | **Alternate GPI SMI Status** — R/WC. These bits report the status of the corresponding GPIOs.<br>0 = Inactive. Software clears this bit by writing a 1 to it.<br>1 = Active<br>These bits are sticky. If the following conditions are true, then an SMI# will be generated and the GPE0_STS bit set:<br>• The corresponding bit in the ALT_GPI_SMI_EN register (PMBASE + 38h) is set<br>• The corresponding GPIO must be routed in the GPI_ROUT register to cause an SMI.<br>• The corresponding GPIO must be implemented.<br>All bits are in the resume well. Default for these bits is dependent on the state of the GPIO pins. |

### 13.8.3.16 GPE_CNTL— General Purpose Control Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE +42h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Usage: | ACPI or Legacy |
| Power Well: | Resume | | |

| Bit | Description |
|---|---|
| 7:2 | **Reserved** |
| 1 | SWGPE_CTRL— R/W. This bit allows software to control the assertion of SWGPE_STS bit. This bit is used by hardware as the level input signal for the SWGPE_STS bit in the GPE0_STS register. When SWGPE_CTRL is 1, SWGPE_STS will be set to 1, and writes to SWGPE_STS with a value of 1 to clear SWGPE_STS will result in SWGPE_STS being set back to 1 by hardware. When SWGPE_CTRL is 0, writes to SWGPE_STS with a value of 1 will clear SWGPE_STS to 0. |
| 0 | THRM#_POL — R/W. This bit controls the polarity of the THRM# pin needed to set the THRM_STS bit.<br>0 = Low value on the THRM# signal will set the THRM_STS bit.<br>1 = HIGH value on the THRM# signal will set the THRM_STS bit. |

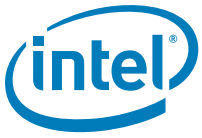
### 13.8.3.17 DEVACT_STS — Device Activity Status Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE +44h | Attribute: | R/WC |
| Default Value: | 0000h | Size: | 16-bit |
| Lockable: | No | Usage: | Legacy Only |
| Power Well: | Core | | |

Each bit indicates if an access has occurred to the corresponding device's trap range, or for bits 6:9 if the corresponding PCI interrupt is active. This register is used in conjunction with the Periodic SMI# timer to detect any system activity for legacy power management. The periodic SMI# timer indicates if it is the right time to read the DEVACT_STS register (PMBASE + 44h).

***Note:*** Software clears bits that are set in this register by writing a 1 to the bit position.

| Bit | Description |
|---|---|
| 15:13 | **Reserved** |
| 12 | **KBC_ACT_STS** — R/WC. KBC (60/64h).<br>0 = Indicates that there has been no access to this device's I/O range.<br>1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location. |
| 11:10 | **Reserved** |
| 9 | **PIRQDH_ACT_STS** — R/WC. PIRQ[D or H].<br>0 = The corresponding PCI interrupts have not been active.<br>1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location. |
| 8 | **PIRQCG_ACT_STS** — R/WC. PIRQ[C or G].<br>0 = The corresponding PCI interrupts have not been active.<br>1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location. |



| Bit | Description |
|---|---|
| 7 | **PIRQBF_ACT_STS** — R/WC. PIRQ[B or F].<br>0 = The corresponding PCI interrupts have not been active.<br>1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location. |
| 6 | **PIRQAE_ACT_STS** — R/WC. PIRQ[A or E].<br>0 = The corresponding PCI interrupts have not been active.<br>1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location. |
| 5:1 | **Reserved** |
| 0 | **IDE_ACT_STS** — R/WC. IDE Primary Drive 0 and Drive 1.<br>0 = Indicates that there has been no access to this device's I/O range.<br>1 = This device's I/O range has been accessed. The enable bit is in the ATC register (D31:F1:Offset C0h). Clear this bit by writing a 1 to the bit location. |

### 13.8.3.18 SS_CNT— Intel SpeedStep® Technology Control Register (Netbook Only)

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE +50h | Attribute: | R/W (special) |
| Default Value | 01h | Size: | 8-bit |
| Lockable: | No | Usage: | ACPI/Legacy |
| Power Well: | Core | | |

***Note:*** Writes to this register will initiate an Intel SpeedStep technology transition that involves a temporary transition to a C3-like state in which the STPCLK# signal will go active. An Intel SpeedStep technology transition **always** occur on writes to the SS_CNT register, even if the value written to SS_STATE is the same as the previous value (after this "transition" the system would still be in the same Intel SpeedStep technology state). If the SS_EN bit is 0, then writes to this register will have no effect and reads will return 0.

| Bit | Description |
|---|---|
| 7:1 | **Reserved** |
| 0 | **SS_STATE** (Intel SpeedStep® technology State) — R/W (Special). When this bit is read, it returns the last value written to this register. By convention, this will be the current Intel SpeedStep technology state. Writes to this register causes a change to the Intel SpeedStep technology state indicated by the value written to this bit. If the new value for SS_STATE is the same as the previous value, then transition will still occur.<br>0 = High power state.<br>1 = Low power state<br><br>**NOTE:** This is only a convention because the transition is the same regardless of the value written to this bit. |

#### 13.8.3.19 C3_RES— C3 Residency Register (Netbook Only)

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE +54h | Attribute: | R/W/RO |
| Default Value | 00000000h | Size: | 32-bit |
| Lockable: | No | Usage: | ACPI/Legacy |
| Power Well: | Core | | |

The value in this field increments at the same rate as the Power Management Timer. This field increments while STP_CPU# is active (i.e. the CPU is in a C3 or C4 state). This field will roll over in the same way as the Power Management Timer, however the most significant bit is NOT sticky.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** |
| 23:0 | C3_RESIDENCY — RO. The value in this field increments at the same rate as the Power Management Timer. If the C3_RESEDENCY_MODE bit is clear, this field automatically resets to 0 at the point when the Lvl3 or Lvl4 read occurs. If the C3_RESIDENCY_MODE bit is set, the register does not reset when the Lvl3 or Lvl4 read occurs. In either mode, it increments while STP_CPU# is active (i.e., the processor is in a C3 or C4 state). This field will roll over in the same way as the PM Timer, however the most significant bit is NOT sticky.<br>Software is responsible for reading this field before performing the Lvl3/4 transition. Software must also check for rollover if the maximum time in C3/C4 could be exceeded.<br><br>**NOTE:** Hardware reset is the only reset of this counter field. |

## 13.9 System Management TCO Registers (D31:F0)

The TCO logic is accessed via registers mapped to the PCI configuration space (Device 31:Function 0) and the system I/O space. For TCO PCI Configuration registers, see LPC Device 31:Function 0 PCI Configuration registers.

### TCO Register I/O Map

The TCO I/O registers reside in a 32-byte range pointed to by a TCOBASE value, which is, PMBASE + 60h in the PCI configuration space. The following table shows the mapping of the registers within that 32-byte range. Each register is described in the following sections.

**Table 13-130.TCO I/O Register Address Map**

| TCOBASE + Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00h–01h | TCO_RLD | TCO Timer Reload and Current Value | 0000h | R/W |
| 02h | TCO_DAT_IN | TCO Data In | 00h | R/W |
| 03h | TCO_DAT_OUT | TCO Data Out | 00h | R/W |
| 04h–05h | TCO1_STS | TCO1 Status | 0000h | R/WC, RO |
| 06h–07h | TCO2_STS | TCO2 Status | 0000h | R/W, R/WC |

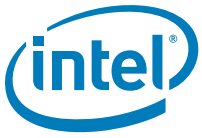


**Table 13-130.TCO I/O Register Address Map**

| TCOBASE + Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 08h–09h | TCO1_CNT | TCO1 Control | 0000h | R/W, R/W (special), R/WC |
| 0Ah–0Bh | TCO2_CNT | TCO2 Control | 0008h | R/W |
| 0Ch–0Dh | TCO_MESSAGE1, TCO_MESSAGE2 | TCO Message 1 and 2 | 00h | R/W |
| 0Eh | TCO_WDCNT | Watchdog Control | 00h | R/W |
| 0Fh | — | **Reserved** | — | — |
| 10h | SW_IRQ_GEN | Software IRQ Generation | 11h | R/W |
| 11h | — | **Reserved** | — | — |
| 12h–13h | TCO_TMR | TCO Timer Initial Value | 0004h | R/W |
| 14h–1Fh | — | **Reserved** | — | — |

## 13.9.1 TCO_RLD—TCO Timer Reload and Current Value Register

| | | | |
|---|---|---|---|
| I/O Address: | TCOBASE +00h | Attribute: | R/W |
| Default Value: | 0000h | Size: | 16-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:10 | **Reserved** |
| 9:0 | **TCO Timer Value** — R/W. Reading this register will return the current count of the TCO timer. Writing any value to this register will reload the timer to prevent the timeout. |

## 13.9.2 TCO_DAT_IN—TCO Data In Register

| | | | |
|---|---|---|---|
| I/O Address: | TCOBASE +02h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | **TCO Data In Value** — R/W. This data register field is used for passing commands from the OS to the SMI handler. Writes to this register will cause an SMI and set the SW_TCO_SMI bit in the TCO1_STS register (D31:F0:04h). |

### 13.9.3 TCO_DAT_OUT—TCO Data Out Register

| | | | |
|---|---|---|---|
| I/O Address: | TCOBASE +03h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | **TCO Data Out Value** — R/W. This data register field is used for passing commands from the SMI handler to the OS. Writes to this register will set the TCO_INT_STS bit in the TCO_STS register. It will also cause an interrupt, as selected by the TCO_INT_SEL bits. |

### 13.9.4 TCO1_STS—TCO1 Status Register

| | | | |
|---|---|---|---|
| I/O Address: | TCOBASE +04h | Attribute: | R/WC, RO |
| Default Value: | 0000h | Size: | 16-bit |
| Lockable: | No | Power Well: | Core<br>(Except bit 7, in RTC) |

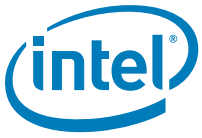
| Bit | Description |
|---|---|
| 15:13 | **Reserved** |
| 12 | **DMISERR_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Chipset received a DMI special cycle message via DMI indicating that it wants to cause an SERR#. The software must read the (G)MCH/CPU to determine the reason for the SERR#. |
| 11 | **Reserved** |
| 10 | **DMISMI_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Chipset received a DMI special cycle message via DMI indicating that it wants to cause an SMI. The software must read the (G)MCH/CPU to determine the reason for the SMI. |
| 9 | **DMISCI_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Chipset received a DMI special cycle message via DMI indicating that it wants to cause an SCI. The software must read the (G)MCH/CPU to determine the reason for the SCI. |
| 8 | **BIOSWR_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Chipset sets this bit and generates and SMI# to indicate an invalid attempt to write to the BIOS. This occurs when either:<br>a) The BIOSWP bit is changed from 0 to 1 and the BLD bit is also set, or<br>b) any write is attempted to the BIOS and the BIOSWP bit is also set.<br><br>**NOTE:** On write cycles attempted to the 4 MB lower alias to the BIOS space, the BIOSWR_STS will not be set. |

| Bit | Description |
|---|---|
| 7 | **NEWCENTURY_STS** — R/WC. This bit is in the RTC well.<br>0 = Cleared by writing a 1 to the bit position or by RTCRST# going active.<br>1 = This bit is set when the Year byte (RTC I/O space, index offset 09h) rolls over from 99 to 00. Setting this bit will cause an SMI# (but not a wake event).<br><br>**NOTE:** The NEWCENTURY_STS bit is not valid when the RTC battery is first installed (or when RTC power has not been maintained). Software can determine if RTC power has not been maintained by checking the RTC_PWR_STS bit (D31:F0:A4h, bit 2), or by other means (such as a checksum on RTC RAM). If RTC power is determined to have not been maintained, BIOS should set the time to a legal value and then clear the NEWCENTURY_STS bit.<br>The NEWCENTURY_STS bit may take up to 3 RTC clocks for the bit to be cleared after a 1 is written to the bit to clear it. After writing a 1 to this bit, software should not exit the SMI handler until verifying that the bit has actually been cleared. This will ensure that the SMI is not re-entered. |
| 6:4 | **Reserved** |
| 3 | **TIMEOUT** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Set by Chipset to indicate that the SMI was caused by the TCO timer reaching 0. |
| 2 | **TCO_INT_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = SMI handler caused the interrupt by writing to the TCO_DAT_OUT register (TCOBASE + 03h). |
| 1 | **SW_TCO_SMI** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Software caused an SMI# by writing to the TCO_DAT_IN register (TCOBASE + 02h). |
| 0 | **NMI2SMI_STS** — RO.<br>0 = Cleared by clearing the associated NMI status bit.<br>1 = Set by the Chipset when an SMI# occurs because an event occurred that would otherwise have caused an NMI (because NMI2SMI_EN is set). |

### 13.9.5 TCO2_STS—TCO2 Status Register

| | | | |
|---|---|---|---|
| I/O Address: | TCOBASE +06h | Attribute: | R/W, R/WC |
| Default Value: | 0000h | Size: | 16-bit |
| Lockable: | No | Power Well: | Resume (Except Bit 0, in RTC) |

| Bit | Description |
|---|---|
| 15:5 | **Reserved** |
| 4 | **SMLink Slave SMI Status (SMLINK_SLV_SMI_STS)** — R/WC. Allow the software to go directly into pre-determined sleep state. This avoids race conditions. Software clears this bit by writing a 1 to it.<br>0 = The bit is reset by RSMRST#, but not due to the PCI Reset associated with exit from S3–S5 states.<br>1 = Chipset sets this bit to 1 when it receives the SMI message on the SMLink's Slave Interface. |
| 3 | **Reserved** |
| 2 | **BOOT_STS** — R/WC.<br>0 = Cleared by Chipset based on RSMRST# or by software writing a 1 to this bit. Note that software should first clear the SECOND_TO_STS bit before writing a 1 to clear the BOOT_STS bit.<br>1 = Set to 1 when the SECOND_TO_STS bit goes from 0 to 1 and the processor has not fetched the first instruction.<br>If rebooting due to a second TCO timer timeout, and if the BOOT_STS bit is set, the Chipset will reboot using the 'safe' multiplier (1111). This allows the system to recover from a processor frequency multiplier that is too high, and allows the BIOS to check the BOOT_STS bit at boot. If the bit is set and the frequency multiplier is 1111, then the BIOS knows that the processor has been programmed to an invalid multiplier. |
| 1 | **SECOND_TO_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it, or by a RSMRST#.<br>1 = Chipset sets this bit to 1 to indicate that the TIMEOUT bit had been (or is currently) set and a second timeout occurred before the TCO_RLD register was written. If this bit is set and the NO_REBOOT config bit is 0, then the Chipset will reboot the system after the second timeout. The reboot is done by asserting PLTRST#. |
| 0 | **Intruder Detect (INTRD_DET)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it, or by RTCRST# assertion.<br>1 = Set by Chipset to indicate that an intrusion was detected. This bit is set even if the system is in G3 state.<br><br>**NOTE:** This bit has a recovery time. After writing a 1 to this bit position (to clear it), the bit may be read back as a 1 for up 65 microseconds before it is read as a 0. Software must be aware of this recovery time when reading this bit after clearing it.<br>**NOTE:** If the INTRUDER# signal is active when the software attempts to clear the INTRD_DET bit, the bit will remain as a 1, and the SMI# will be generated again immediately. The SMI handler can clear the INTRD_SEL bits (TCOBASE + 0Ah, bits 2:1), to avoid further SMIs. However, if the INTRUDER# signals goes inactive and then active again, there will not be further SMI's (because the INTRD_SEL bits would select that no SMI# be generated).<br>**NOTE:** If the INTRUDER# signal goes inactive some point after the INTRD_DET bit is written as a 1, then the INTRD_DET signal will go to a 0 when INTRUDER# input signal goes inactive. Note that 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. |



## 13.9.6 TCO1_CNT—TCO1 Control Register

I/O Address: TCOBASE +08h
Default Value: 0000h
Lockable: No
Attribute:R/W, R/W (special), R/WC
Size:16-bit
Power Well:Core

| Bit | Description |
|---|---|
| 15:13 | **Reserved** |
| 12 | **TCO_LOCK** — R/W (special). When set to 1, this bit prevents writes from changing the TCO_EN bit (in offset 30h of Power Management I/O space). Once this bit is set to 1, it can not be cleared by software writing a 0 to this bit location. A core-well reset is required to change this bit from 1 to 0. This bit defaults to 0. |
| 11 | **TCO Timer Halt (TCO_TMR_HLT)** — R/W.<br>0 = The TCO Timer is enabled to count.<br>1 = The TCO Timer will halt. It will not count, and thus cannot reach a value that will cause an SMI# or set the SECOND_TO_STS bit. When set, this bit will prevent rebooting and prevent Alert On LAN event messages from being transmitted on the SMLINK (but not Alert On LAN* heartbeat messages). |
| 10 | **SEND_NOW** — R/W (special).<br>0 = The Chipset will clear this bit when it has completed sending the message. Software must not set this bit to 1 again until the Chipset has set it back to 0.<br>1 = Chipset sends an Alert On LAN Event message over the SMLINK interface, with the Software Event bit set.<br>Setting the SEND_NOW bit causes the Chipset integrated LAN controller to reset, which can have unpredictable side-effects. Unless software protects against these side effects, software should not attempt to set this bit. |
| 9 | **NMI2SMI_EN** — R/W.<br>0 = Normal NMI functionality.<br>1 = Forces all NMIs to instead cause SMIs. The functionality of this bit is dependent upon the settings of the NMI_EN bit and the GBL_SMI_EN bit as detailed in the following table: |
| 8 | **NMI_NOW** — R/WC.<br>0 = Software clears this bit by writing a 1 to it. The NMI handler is expected to clear this bit. Another NMI will not be generated until the bit is cleared.<br>1 = Writing a 1 to this bit causes an NMI. This allows the BIOS or SMI handler to force an entry to the NMI handler. |
| 7:0 | **Reserved** |

| NMI_EN | GBL_SMI_EN | Description |
|---|---|---|
| 0b | 0b | No SMI# at all because GBL_SMI_EN = 0 |
| 0b | 1b | SMI# will be caused due to NMI events |
| 1b | 0b | No SMI# at all because GBL_SMI_EN = 0 |
| 1b | 1b | No SMI# due to NMI because NMI_EN = 1 |

## 13.9.7 TCO2_CNT—TCO2 Control Register

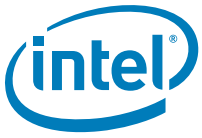
| | | | |
|---|---|---|---|
| I/O Address: | TCOBASE +0Ah | Attribute: | R/W |
| Default Value: | 0008h | Size: | 16-bit |
| Lockable: | No | Power Well: | Resume |

| Bit | Description |
|---|---|
| 15:6 | **Reserved** |
| 5:4 | **OS_POLICY** — R/W. OS-based software writes to these bits to select the policy that the BIOS will use after the platform resets due the WDT. The following convention is recommended for the BIOS and OS:<br>00 = Boot normally<br>01 = Shut down<br>10 = Don't load OS. Hold in pre-boot state and use LAN to determine next step<br>11 = **Reserved**<br><br>**NOTE:** These are just scratchpad bits. They should not be reset when the TCO logic resets the platform due to Watchdog Timer. |
| 3 | **GPIO11_ALERT_DISABLE** — R/W. At reset (via RSMRST# asserted) this bit is set and GPIO11 alerts are disabled.<br>0 = Enable.<br>1 = Disable GPIO11/SMBALERT# as an alert source for the heartbeats and the SMBus slave. |
| 2:1 | **INTRD_SEL** — R/W. This field selects the action to take if the INTRUDER# signal goes active.<br>00 = No interrupt or SMI#<br>01 = Interrupt (as selected by TCO_INT_SEL).<br>10 = SMI<br>11 = **Reserved** |
| 0 | **Reserved** |

## 13.9.8 TCO_MESSAGE1 and TCO_MESSAGE2 Registers

| | | | |
|---|---|---|---|
| I/O Address: | TCOBASE +0Ch (Message 1)<br>TCOBASE +0Dh (Message 2) | Attribute: | R/W |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Power Well: | Resume |

| Bit | Description |
|---|---|
| 7:0 | **TCO_MESSAGE[*n*]** — R/W. The value written into this register will be sent out via the SMLINK interface in the MESSAGE field of the Alert On LAN message. BIOS can write to this register to indicate its boot progress which can be monitored externally. |



### 13.9.9 TCO_WDCNT—TCO Watchdog Control Register

| | | | |
|---|---|---|---|
| Offset Address: | TCOBASE + 0Eh | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bits |
| Power Well: | Resume | | |

| Bit | Description |
|---|---|
| 7:0 | **Watchdog Status (WDSTATUS)** — R/W. The value written to this register will be sent in the Alert On LAN message on the SMLINK interface. It can be used by the BIOS or system management software to indicate more details on the boot progress. This register will be reset to the default of 00h based on RSMRST# (but not PCI reset). |

### 13.9.10 SW_IRQ_GEN—Software IRQ Generation Register

| | | | |
|---|---|---|---|
| Offset Address: | TCOBASE + 10h | Attribute: | R/W |
| Default Value: | 11h | Size: | 8 bits |
| Power Well: | Core | | |

| Bit | Description |
|---|---|
| 7:2 | **Reserved** |
| 1 | **IRQ12_CAUSE** — R/W. The state of this bit is logically ANDed with the IRQ12 signal as received by the chipset's SERIRQ logic. This bit must be a 1 (default) if the Chipset is expected to receive IRQ12 assertions from a SERIRQ device. |
| 0 | **IRQ1_CAUSE** — R/W. The state of this bit is logically ANDed with the IRQ1 signal as received by the chipset's SERIRQ logic. This bit must be a 1 (default) if the Chipset is expected to receive IRQ1 assertions from a SERIRQ device. |

### 13.9.11 TCO_TMR—TCO Timer Initial Value Register

| | | | |
|---|---|---|---|
| I/O Address: | TCOBASE +12h | Attribute: | R/W |
| Default Value: | 0004h | Size: | 16-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:10 | **Reserved** |
| 9:0 | **TCO Timer Initial Value** — R/W. Value that is loaded into the timer each time the TCO_RLD register is written. Values of 0000h or 0001h will be ignored and should not be attempted. The timer is clocked at approximately 0.6 seconds, and thus allows timeouts ranging from 1.2 second to 613.8 seconds. Note: The timer has an error of ± 1 tick (0.6s).<br>The TCO Timer will only count down in the S0 state. |

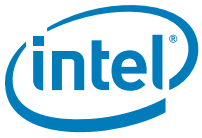
## 13.10 General Purpose I/O Registers (D31:F0)

The control for the general purpose I/O signals is handled through a separate 64-byte I/O space. The base offset for this space is selected by the GPIOBASE register.

GPIO Register I/O Address Map

**Table 13-131.Registers to Control GPIO Address Map**

| GPIOBASE + Offset | Mnemonic | Register Name | Default | Access |
|---|---|---|---|---|
| **General Registers** | | | | |
| 00h–03h | GPIO_USE_SEL | GPIO Use Select | 1F2AF7FFh | R/W |
| 04h–07h | GP_IO_SEL | GPIO Input/Output Select | E0E8FFFFh | R/W |
| 08h–0Bh | — | **Reserved** | — | — |
| 0Ch–0Fh | GP_LVL | GPIO Level for Input or Output | 02FE0000h | R/W |
| 10h–13h | | **Reserved** | — | — |
| **Output Control Registers** | | | | |
| 14h–17h | — | **Reserved** | — | — |
| 18h–1Bh | GPO_BLINK | GPIO Blink Enable | 00040000h | R/W |
| 1Ch–1Fh | — | **Reserved** | — | — |
| **Input Control Registers** | | | | |
| 20–2Bh | — | **Reserved** | — | — |
| 2C–2Fh | GPI_INV | GPIO Signal Invert | 00000000h | R/W |
| 30h–33h | GPIO_USE_SEL2 | GPIO Use Select 2 [63:32] | 000300FEh | R/W |
| 34h–37h | GP_IO_SEL2 | GPIO Input/Output Select 2 [63:32] | 000000F0h | R/W |
| 38h–3Bh | GP_LVL2 | GPIO Level for Input or Output 2 [63:32] | 00030003h | R/W |



### 13.10.1 GPIO_USE_SEL—GPIO Use Select Register

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE + 00h | Attribute: | R/W |
| Default Value: | 1F2AF7FFh | Size: | 32-bit |
| Lockable: | No | Power Well: | Core for 0:7, 16:23, Resume for 8:15, 24:31 |

| Bit | Description |
|---|---|
| 31:0 | **GPIO_USE_SEL[31:0]** — R/W. Each bit in this register enables the corresponding GPIO (if it exists) to be used as a GPIO, rather than for the native function.<br>0 = Signal used as native function.<br>1 = Signal used as a GPIO.<br><br>**NOTES:**<br>1. The following bits are always 1 because they are unmuxed: 6:10,12:15, 24:25<br>2. The following bits are not implemented because they are determined by the configuration: 16, 18, 20, 32<br>3. If GPIO[n] does not exist, then the bit in this register will always read as 0 and writes will have no effect.<br>4. After a full reset (RSMRST#) all multiplexed signals in the resume and core wells are configured as their default function. After just a PLTRST#, the GPIO in the core well are configured as their default function.<br>5. When configured to GPIO mode, the multiplexing logic will present the inactive state to native logic that uses the pin as an input.<br>6. All GPIOs are reset to the default state by CF9h reset except GPIO24 |

### 13.10.2 GP_IO_SEL—GPIO Input/Output Select Register

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +04h | Attribute: | R/W |
| Default Value: | E0E8FFFFh | Size: | 32-bit |
| Lockable: | No | Power Well: | Core for 0:7, 16:23, Resume for 8:15, 24:31 |

| Bit | Description |
|---|---|
| 31:0 | **GP_IO_SEL[31:0]** — R/W. When configured in native mode (GPIO_USE_SEL[n] is 0), writes to these bits have no effect. The value reported in this register is undefined when programmed as native mode.<br>0 = Output. The corresponding GPIO signal is an output.<br>1 = Input. The corresponding GPIO signal is an input. |

### 13.10.3 GP_LVL—GPIO Level for Input or Output Register

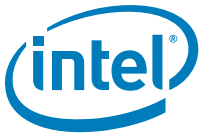
| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +0Ch | Attribute: | R/W |
| Default Value: | 02FE0000h | Size: | 32-bit |
| Lockable: | No | Power Well: | Core for 0:7, 16:23, Resume for 8:15, 24:31 |

| Bit | Description |
|---|---|
| 31:0 | **GP_LVL[31:0]**— R/W: If GPIO[n] is programmed to be an output (via the corresponding bit in the GP_IO_SEL register), then the corresponding GP_LVL[n] bit can be updated by software to drive a high or low value on the output pin. 1 = high, 0 = low.<br>If GPIO[n] is programmed as an input, then the corresponding GP_LVL bit reflects the state of the input signal (1 = high, 0 = low.) and writes will have no effect.<br>When configured in native mode (GPIO_USE_SEL[n] is 0), writes to these bits have no effect. The value reported in this register is undefined when programmed as native mode. |

### 13.10.4 GPO_BLINK—GPO Blink Enable Register

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +18h | Attribute: | R/W |
| Default Value: | 00040000h | Size: | 32-bit |
| Lockable: | No | Power Well: | Core for 0:7, 16:23, Resume for 8:15, 24:31 |

| Bit | Description |
|---|---|
| 31:0 | **GP_BLINK[31:0]** — R/W. The setting of this bit has no effect if the corresponding GPIO signal is programmed as an input.<br>0 = The corresponding GPIO will function normally.<br>1 = If the corresponding GPIO is programmed as an output, the output signal will blink at a rate of approximately once per second. The high and low times have approximately 0.5 seconds each. The GP_LVL bit is not altered when this bit is set.<br>The value of the corresponding GP_LVL bit remains unchanged during the blink process, and does not effect the blink in any way. The GP_LVL bit is not altered when programmed to blink. It will remain at its previous value.<br>These bits correspond to GPIO in the Resume well. These bits revert to the default value based on RSMRST# or a write to the CF9h register (but not just on PLTRST#). |



## 13.10.5 GPI_INV—GPIO Signal Invert Register

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +2Ch | Attribute: | R/W |
| Default Value: | 00000000h | Size: | 32-bit |
| Lockable: | No | Power Well: | Core for 0:7, 16:23, Resume for 8:15, 24:31 |

| Bit | Description |
|---|---|
| 31:0 | **GP_INV[n]** — R/W. **Input Inversion:** This bit only has effect if the corresponding GPIO is used as an input and used by the GPE logic, where the polarity matters. When set to 1, then the GPI is inverted as it is sent to the GPE logic that is using it. This bit has no effect on the value that is reported in the GP_LVL register.<br>These bits are used to allow both active-low and active-high inputs to cause SMI# or SCI. Note that in the S0 or S1 state, the input signal must be active for at least two PCI clocks to ensure detection by the Chipset. In the S3, S4 or S5 states the input signal must be active for at least 2 RTC clocks to ensure detection. The setting of these bits has no effect if the corresponding GPIO is programmed as an output. These bits correspond to GPI that are in the resume well, and will be reset to their default values by RSMRST# or by a write to the CF9h register.<br>0 = The corresponding GPI_STS bit is set when the Chipset detects the state of the input pin to be high.<br>1 = The corresponding GPI_STS bit is set when the Chipset detects the state of the input pin to be low. |

## 13.10.6 GPIO_USE_SEL2—GPIO Use Select 2 Register[63:32]

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +30h | Attribute:R/W | |
| Default Value: | 000300FEh | Size:32-bit | |
| Lockable: | No | Power Well:CPU I/O for 17, Core for 16, 7:0 | |

| Bit | Description |
|---|---|
| 17:16, 7:0 | **GPIO_USE_SEL2[49:48, 39:32] Bits[17:16, 7:0]**— R/W. Each bit in this register enables the corresponding GPIO (if it exists) to be used as a GPIO, rather than for the native function.<br>0 = Signal used as native function.<br>1 = Signal used as a GPIO.<br>After a full reset (RSMRST#), all multiplexed signals in the resume and core wells are configured as a GPIO rather than as their native function. After just a PLTRST#, the GPIO in the core well are configured as GPIO.<br><br>**NOTES:**<br>1. The following bits are not implemented because there is no corresponding GPIO: 31:18, 15:8.<br>2. The following bits are not implemented because they are determined by the configuration: 0 |

### 13.10.7 GP_IO_SEL2—GPIO Input/Output Select 2 Register[63:32]

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +34h | Attribute: | R/W |
| Default Value: | 000000F0h | Size: | 32-bit |
| Lockable: | No | Power Well: | CPU I/O for 17, Core for 16, 7:0 |

| Bit | Description |
|---|---|
| 31:18, 15:8 | Always 0. No corresponding GPIO. |
| 17:16, 7:0 | **GP_IO_SEL2[49:48, 39:32]** — R/W.<br>0 = GPIO signal is programmed as an output.<br>1 = Corresponding GPIO signal (if enabled in the GPIO_USE_SEL2 register) is programmed as an input. |

### 13.10.8 GP_LVL2—GPIO Level for Input or Output 2 Register[63:32]

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +38h | Attribute: | R/W |
| Default Value: | 00030003h | Size: | 32-bit |
| Lockable: | No | Power Well: | CPU I/O for 17, Core for 16:0 |

| Bit | Description |
|---|---|
| 31:18, 15:8 | **Reserved**. Read-only 0 |
| 17:16, 7:0 | **GP_LVL[49:48, 39:32]** — R/W.<br>If GPIO[n] is programmed to be an output (via the corresponding bit in the GP_IO_SEL register), then the corresponding GP_LVL[n] bit can be updated by software to drive a high or low value on the output pin. 1 = high, 0 = low.<br>If GPIO[n] is programmed as an input, then the corresponding GP_LVL bit reflects the state of the input signal (1 = high, 0 = low.) and writes will have no effect.<br>When configured in native mode (GPIO_USE_SEL[n] is 0), writes to these bits have no effect. The value reported in this register is undefined when programmed as native mode. |

§

# 14 UHCI Controllers Registers

## 14.1 PCI Configuration Registers (USB—D29:F0/F1/F2/F3)

***Note:*** Register address locations that are not shown in Table 14-132 and should be treated as **Reserved** (see Section 9.2 for details).

**Table 14-132.UHCI Controller PCI Register Address Map (USB—D29:F0/F1/F2/F3)**

| Offset | Mnemonic | Register Name | Function 0 Default | Function 1 Default | Function 2 Default | Function 3 Default | Type |
|---|---|---|---|---|---|---|---|
| 00–01h | VID | Vendor Identification | 8086h | 8086h | 8086h | 8086h | RO |
| 02–03h | DID | Device Identification | See register description | See register description | See register description | See register description | RO |
| 04–05h | PCICMD | PCI Command | 0000h | 0000h | 0000h | 0000h | R/W, RO |
| 06–07h | PCISTS | PCI Status | 0280h | 0280h | 0280h | 0280h | R/WC, RO |
| 08h | RID | Revision Identification | See register description | See register description | See register description | See register description | RO |
| 09h | PI | Programming Interface | 00h | 00h | 00h | 00h | RO |
| 0Ah | SCC | Sub Class Code | 03h | 03h | 03h | 03h | RO |
| 0Bh | BCC | Base Class Code | 0Ch | 0Ch | 0Ch | 0Ch | RO |
| 0Dh | MLT | Master Latency Timer | 00h | 00h | 00h | 00h | RO |
| 0Eh | HEADTYP | Header Type | 80h | 00h | 00h | 00h | RO |
| 20–23h | BASE | Base Address | 00000001h | 00000001h | 00000001h | 00000001h | R/W, RO |
| 2C–2Dh | SVID | Subsystem Vendor Identification | 0000h | 0000h | 0000h | 0000h | R/WO |
| 2E–2Fh | SID | Subsystem Identification | 0000h | 0000h | 0000h | 0000h | R/WO |
| 3Ch | INT_LN | Interrupt Line | 00h | 00h | 00h | 00h | R/W |
| 3Dh | INT_PN | Interrupt Pin | See register description. | See register description. | See register description. | See register description. | RO |
| 60h | USB_RELNUM | Serial Bus Release Number | 10h | 10h | 10h | 10h | RO |
| C0–C1h | USB_LEGKEY | USB Legacy Keyboard/Mouse Control | 2000h | 2000h | 2000h | 2000h | R/W, RO R/WC |
| C4h | USB_RES | USB Resume Enable | 00h | 00h | 00h | 00h | R/W |
| C8h | CWP | Core Well Policy | 00h | 00h | 00h | 00h | R/W |

### 14.1.1 VID—Vendor Identification Register (USB—D29:F0/F1/F2/F3)

Address Offset: 00h–01h Attribute: RO

Default Value: 8086hSize:16 bits

| Bit | Description |
|---|---|
| 15:0 | **Vendor ID** — RO. This is a 16-bit value assigned to Intel |

### 14.1.2 DID—Device Identification Register (USB—D29:F0/F1/F2/F3)

Address Offset: 02h–03hAttribute:RO

Default Value: UHCI #1 = See bit descriptionSize:16 bits

UHCI #2 = See bit description

UHCI #3 = See bit description

UHCI #4 = See bit description

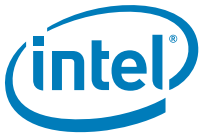
| Bit | Description |
|---|---|
| 15:0 | **Device ID** — RO. This is a 16-bit value assigned to the Chipset USB universal host controllers. |

### 14.1.3 PCICMD—PCI Command Register (USB—D29:F0/F1/F2/F3)

Address Offset: 04h–05hAttribute:R/W, RO

Default Value: 0000hSize:16 bits

| Bit | Description |
|---|---|
| 15:11 | **Reserved** |
| 10 | **Interrupt Disable** — R/W.<br>0 = Enable. The function is able to generate its interrupt to the interrupt controller.<br>1 = Disable. The function is not capable of generating interrupts.<br><br>**NOTE:** The corresponding Interrupt Status bit is not affected by the interrupt enable. |
| 9 | **Fast Back to Back Enable (FBE)** — RO. Hardwired to 0. |
| 8 | **SERR# Enable** — RO. **Reserved** as 0. |
| 7 | **Wait Cycle Control (WCC)** — RO. Hardwired to 0. |
| 6 | **Parity Error Response (PER)** — RO. Hardwired to 0. |
| 5 | **VGA Palette Snoop (VPS)** — RO. Hardwired to 0. |
| 4 | **Postable Memory Write Enable (PMWE)** — RO. Hardwired to 0. |



| Bit | Description |
|---|---|
| 3 | **Special Cycle Enable (SCE)** — RO. Hardwired to 0. |
| 2 | **Bus Master Enable (BME)** — R/W.<br>0 = Disable<br>1 = Enable. Chipset can act as a master on the PCI bus for USB transfers. |
| 1 | Memory Space Enable (MSE) — RO. Hardwired to 0. |
| 0 | **I/O Space Enable (IOSE)** — R/W. This bit controls access to the I/O space registers.<br>0 = Disable<br>1 = Enable accesses to the USB I/O registers. The Base Address register for USB should be programmed before this bit is set. |

### 14.1.4 PCISTS—PCI Status Register (USB—D29:F0/F1/F2/F3)

Address Offset: 06h–07hAttribute:R/WC, RO

Default Value: 0280hSize:16 bits

***Note:*** For the writable bits, software must write a 1 to clear bits that are set. Writing a 0 to the bit has no effect.

| Bit | Description |
|---|---|
| 15 | **Detected Parity Error (DPE)** — R/WC.<br>0 = No parity error detected.<br>1 = Set when a data parity error data parity error is detected on writes to the UHCI register space or on read completions returned to the host controller. |
| 14 | **Reserved** as 0b. Read Only. |
| 13 | **Received Master Abort (RMA)** — R/WC.<br>0 = No master abort generated by USB.<br>1 = USB, as a master, generated a master abort. |
| 12 | **Reserved**. Always read as 0. |
| 11 | **Signaled Target Abort (STA)** — R/WC.<br>0 = Chipset did Not terminate transaction for USB function with a target abort.<br>1 = USB function is targeted with a transaction that the Chipset terminates with a target abort. |
| 10:9 | **DEVSEL# Timing Status (DEV_STS)** — RO. This 2-bit field defines the timing for DEVSEL# assertion. These read only bits indicate the Chipset's DEVSEL# timing when performing a positive decode. Chipset generates DEVSEL# with medium timing for USB. |
| 8 | **Data Parity Error Detected (DPED)** — RO. Hardwired to 0. |
| 7 | **Fast Back to Back Capable (FB2BC)** — RO. Hardwired to 1. |
| 6 | **User Definable Features (UDF)** — RO. Hardwired to 0. |
| 5 | **66 MHz Capable** — RO. Hardwired to 0. |

| Bit | Description |
|---|---|
| 4 | **Capabilities List — RO**. Hardwired to 0. |
| 3 | **Interrupt Status** — RO. This bit reflects the state of this function's interrupt at the input of the enable/disable logic.<br>0 = Interrupt is de-asserted.<br>1 = Interrupt is asserted.<br>The value reported in this bit is independent of the value in the Interrupt Enable bit. |
| 2:0 | **Reserved** |

### 14.1.5 RID—Revision Identification Register (USB—D29:F0/F1/F2/F3)

Offset Address: 08h Attribute: RO
Default Value: See bit description Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Revision ID** — RO. |

### 14.1.6 PI—Programming Interface Register (USB—D29:F0/F1/F2/F3)

Address Offset: 09h Attribute: RO

Default Value: 00h Size: 8 bits

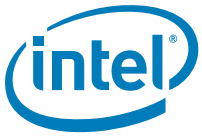
| Bit | Description |
|---|---|
| 7:0 | **Programming Interface** — RO.<br>00h = No specific register level programming interface defined. |

### 14.1.7 SCC—Sub Class Code Register (USB—D29:F0/F1/F2/F3)

Address Offset: 0Ah Attribute: RO

Default Value: 03h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Sub Class Code (SCC)** — RO.<br>03h = USB host controller. |



### 14.1.8 BCC—Base Class Code Register (USB—D29:F0/F1/F2/F3)

Address Offset: 0BhAttribute:RO

Default Value: 0ChSize:8 bits

| Bit | Description |
|---|---|
| 7:0 | Base Class Code (BCC) — RO.<br>0Ch = Serial Bus controller. |

### 14.1.9 MLT—Master Latency Timer Register (USB—D29:F0/F1/F2/F3)

Address Offset: 0Dh Attribute: RO
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Master Latency Timer (MLT) — RO. The USB controller is implemented internal to the Chipset and not arbitrated as a PCI device. Therefore the device does not require a Master Latency Timer. |

### 14.1.10 HEADTYP—Header Type Register (USB—D29:F0/F1/F2/F3)

Address Offset: 0EhAttribute:RO

Default Value: FN 0: 80hSize:8 bits
FN 1: 00h
FN 2: 00h
FN 3: 00h

For functions 1, 2, and 3, this register is hardwired to 00h. For function 0, bit 7 is determined by the values in the USB Function Disable bits (11:8 of the Function Disable register Chipset Config Registers:Offset 3418h).

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7</td><td>Multi-Function Device — RO.<br>0 = Single-function device.<br>1 = Multi-function device.<br>Since the upper functions in this device can be individually hidden, this bit is based on the function-disable bits in Chipset Config Space: Offset 3418h as follows:
<table>
<tr><th>D29:F7_Disable (bit 15)</th><th>D29:F3_Disable (bit 11)</th><th>D29:F2_Disable (bit10)</th><th>D29:F1_Disable (bit 9)</th><th>Multi-Function Device (this bit)</th></tr>
<tr><td>0b</td><td>X</td><td>X</td><td>X</td><td>1</td></tr>
<tr><td>X</td><td>0b</td><td>X</td><td>X</td><td>1</td></tr>
<tr><td>X</td><td>X</td><td>0b</td><td>X</td><td>1</td></tr>
<tr><td>X</td><td>X</td><td>X</td><td>0b</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
</table>
</td></tr>
</table>

| Bit | Description |
|---|---|
| 6:0 | Configuration Layout. Hardwired to 00h, which indicates the standard PCI configuration layout. |

### 14.1.11 BASE—Base Address Register (USB—D29:F0/F1/F2/F3)

Address Offset: 20h–23hAttribute:R/W, RO

Default Value: 00000001hSize:32 bits

| Bit | Description |
|---|---|
| 31:16 | **Reserved** |
| 15:5 | **Base Address** — R/W. Bits [15:5] correspond to I/O address signals AD [15:5], respectively. This gives 32 bytes of relocatable I/O space. |
| 4:1 | **Reserved** |
| 0 | Resource Type Indicator (RTE) — RO. Hardwired to 1 to indicate that the base address field in this register maps to I/O space. |

### 14.1.12 SVID — Subsystem Vendor Identification Register (USB—D29:F0/F1/F2/F3)

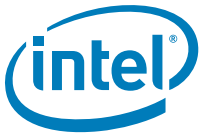
| | | | |
|---|---|---|---|
| Address Offset: | 2Ch–2Dh | Attribute: | R/WO |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:0 | Subsystem Vendor ID (SVID) — R/WO. BIOS sets the value in this register to identify the Subsystem Vendor ID. The USB_SVID register, in combination with the USB Subsystem ID register, enables the operating system to distinguish each subsystem from the others.<br><br>**NOTE:** The software can write to this register only once per core well reset. Writes should be done as a single, 16-bit cycle. |

### 14.1.13 SID — Subsystem Identification Register (USB—D29:F0/F1/F2/F3)

| | | | |
|---|---|---|---|
| Address Offset: | 2Eh–2Fh | Attribute: | R/WO |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:0 | **Subsystem ID (SID)** — R/WO. BIOS sets the value in this register to identify the Subsystem ID. The SID register, in combination with the SVID register (D29:F0/F1/F2/F3:2C), enables the operating system to distinguish each subsystem from other(s). The value read in this register is the same as what was written to the IDE_SID register.<br><br>**NOTE:** The software can write to this register only once per core well reset. Writes should be done as a single, 16-bit cycle. |



### 14.1.14 INT_LN—Interrupt Line Register (USB—D29:F0/F1/F2/F3)

Address Offset: 3ChAttribute:R/W

Default Value: 00hSize:8 bits

| Bit | Description |
|---|---|
| 7:0 | **Interrupt Line (INT_LN)** — RO. This data is not used by the Chipset. It is to communicate to software the interrupt line that the interrupt pin is connected to. |

### 14.1.15 INT_PN—Interrupt Pin Register (USB—D29:F0/F1/F2/F3)

Address Offset: 3DhAttribute:RO

Default Value: Function 0: See DescriptionSize:8 bits

Function 1: See Description

Function 2: See Description

Function 3: See Description

| Bit | Description |
|---|---|
| 7:0 | **Interrupt Line (INT_LN)** — RO. This value tells the software which interrupt pin each USB host controller uses. The upper 4 bits are hardwired to 0000b; the lower 4 bits are determine by the Interrupt Pin default values that are programmed in the memory-mapped configuration space as follows:<br>Function 0 D29IP.U0P (Chipset Config Registers:Offset 3108:bits 3:0)<br>Function 1 D29IP.U1P (Chipset Config Registers:Offset 3108:bits 7:4)<br>Function 2 D29IP.U2P (Chipset Config Registers:Offset 3108:bits 11:8)<br>Function 3 D29IP.U3P (Chipset Config Registers:Offset 3108:bits 15:12)<br><br>**NOTE:** This does not determine the mapping to the PIRQ pins. |

### 14.1.16 USB_RELNUM—Serial Bus Release Number Register (USB—D29:F0/F1/F2/F3)

Address Offset: 60hAttribute:RO

Default Value: 10hSize:8 bits

| Bit | Description |
|---|---|
| 7:0 | **Serial Bus Release Number** — RO.<br>10h = USB controller supports the *USB Specification*, Release 1.0. |

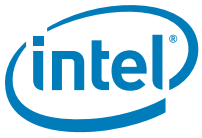
### 14.1.17 USB_LEGKEY—USB Legacy Keyboard/Mouse Control Register (USB—D29:F0/F1/F2/F3)

Address Offset: C0h–C1hAttribute:R/W, R/WC, RO

Default Value: 2000hSize:16 bits

This register is implemented separately in each of the USB UHCI functions. However, the enable and status bits for the trapping logic are OR'd and shared, respectively, since their functionality is not specific to any one host controller.

| Bit | Description |
|---|---|
| 15 | **SMI Caused by End of Pass-Through (SMIBYENDPS)** — R/WC. This bit indicates if the event occurred. Note that even if the corresponding enable bit is not set in bit 7, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#.<br>0 = Software clears this bit by writing a 1 to the bit location in any of the controllers.<br>1 = Event Occurred |
| 14 | **Reserved** |
| 13 | **PCI Interrupt Enable (USBPIRQEN)** — R/W. This bit is used to prevent the USB controller from generating an interrupt due to transactions on its ports. Note that, when disabled, it will probably be configured to generate an SMI using bit 4 of this register. Default to 1 for compatibility with older USB software.<br>0 = Disable<br>1 = Enable |
| 12 | **SMI Caused by USB Interrupt (SMIBYUSB)** — RO. This bit indicates if an interrupt event occurred from this controller. The interrupt from the controller is taken before the enable in bit 13 has any effect to create this read-only bit. Note that even if the corresponding enable bit is not set in Bit 4, this bit may still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#.<br>0 = Software should clear the interrupts via the USB controllers. Writing a 1 to this bit will have no effect.<br>1 = Event Occurred. |
| 11 | **SMI Caused by Port 64 Write (TRAPBY64W)** — R/WC. This bit indicates if the event occurred. Note that even if the corresponding enable bit is not set in bit 3, this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Note that the A20Gate Pass-Through Logic allows specific port 64h writes to complete without setting this bit.<br>0 = Software clears this bit by writing a 1 to the bit location in any of the controllers.<br>1 = Event Occurred. |
| 10 | **SMI Caused by Port 64 Read (TRAPBY64R)** — R/WC. This bit indicates if the event occurred. Note that even if the corresponding enable bit is not set in bit 2, this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#.<br>0 = Software clears this bit by writing a 1 to the bit location in any of the controllers.<br>1 = Event Occurred. |



| Bit | Description |
|---|---|
| 9 | **SMI Caused by Port 60 Write (TRAPBY60W)** — R/WC. This bit indicates if the event occurred. Note that even if the corresponding enable bit is not set in bit 1, this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Note that the A20Gate Pass-Through Logic allows specific port 64h writes to complete without setting this bit.<br>0 = Software clears this bit by writing a 1 to the bit location in any of the controllers.<br>1 = Event Occurred. |
| 8 | **SMI Caused by Port 60 Read (TRAPBY60R)** — R/WC. This bit indicates if the event occurred. Note that even if the corresponding enable bit is not set in the bit 0, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#.<br>0 = Software clears this bit by writing a 1 to the bit location in any of the controllers.<br>1 = Event Occurred. |
| 7 | **SMI at End of Pass-Through Enable (SMIATENDPS)** — R/W. This bit enables SMI at the end of a pass-through. This can occur if an SMI is generated in the middle of a pass-through, and needs to be serviced later.<br>0 = Disable<br>1 = Enable |
| 6 | **Pass Through State (PSTATE)** — RO.<br>0 = If software needs to reset this bit, it should set bit 5 in all of the host controllers to 0.<br>1 = Indicates that the state machine is in the middle of an A20GATE pass-through sequence. |
| 5 | **A20Gate Pass-Through Enable (A20PASSEN)** — R/W.<br>0 = Disable.<br>1 = Enable. Allows A20GATE sequence Pass-Through function. A specific cycle sequence involving writes to port 60h and 64h does not result in the setting of the SMI status bits. |
| 4 | **SMI on USB IRQ Enable (USBSMIEN)** — R/W.<br>0 = Disable<br>1 = Enable. USB interrupt will cause an SMI event. |
| 3 | **SMI on Port 64 Writes Enable (64WEN)** — R/W.<br>0 = Disable<br>1 = Enable. A 1 in bit 11 will cause an SMI event. |
| 2 | **SMI on Port 64 Reads Enable (64REN)** — R/W.<br>0 = Disable<br>1 = Enable. A 1 in bit 10 will cause an SMI event. |
| 1 | **SMI on Port 60 Writes Enable (60WEN)** — R/W.<br>0 = Disable<br>1 = Enable. A 1 in bit 9 will cause an SMI event. |
| 0 | **SMI on Port 60 Reads Enable (60REN)** — R/W.<br>0 = Disable<br>1 = Enable. A 1 in bit 8 will cause an SMI event. |

### 14.1.18 USB_RES—USB Resume Enable Register (USB—D29:F0/F1/F2/F3)

Address Offset: C4hAttribute:R/W

Default Value: 00hSize:8 bits

| Bit | Description |
|---|---|
| 7:2 | **Reserved** |
| 1 | **PORT1EN** — R/W. Enable port 1 of the USB controller to respond to wakeup events.<br>0 = The USB controller will not look at this port for a wakeup event.<br>1 = The USB controller will monitor this port for remote wakeup and connect/ disconnect events. |
| 0 | **PORT0EN** — R/W. Enable port 0 of the USB controller to respond to wakeup events.<br>0 = The USB controller will not look at this port for a wakeup event.<br>1 = The USB controller will monitor this port for remote wakeup and connect/ disconnect events. |

### 14.1.19 CWP—Core Well Policy Register (USB—D29:F0/F1/F2/F3)

Address Offset: C8h Attribute: R/W
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:1 | **Reserved** |
| 0 | **Static Bus Master Status Policy Enable** (SBMSPE) — R/W.<br>0 = The UHCI host controller dynamically sets the Bus Master status bit (Power Management 1 Status Register,[PMBASE+00h], bit 4) based on the memory accesses that are scheduled. For Netbook only, the default setting provides a more accurate indication of snoopable memory accesses in order to help with software-invoked entry to C3 and C4 power states.<br>1 = The UHCI host controller statically forces the Bus Master Status bit in power management space to 1 whenever the HCHalted bit (USB Status Register, Base+02h, bit 5) is cleared.<br><br>**NOTE:** The PCI Power Management registers are enabled in the PCI Device 31: Function 0 space (PM_IO_EN), and can be moved to any I/O location (128-byte aligned). |

## 14.2 USB I/O Registers

Some of the read/write register bits that deal with changing the state of the USB hub ports function such that on read back they reflect the current state of the port, and not necessarily the state of the last write to the register. This allows the software to poll the state of the port and wait until it is in the proper state before proceeding. A host controller reset, global reset, or port reset will immediately terminate a transfer on the affected ports and disable the port. This affects the USBCMD register, bit 4 and the PORTSC registers, bits [12,6,2]. See individual bit descriptions for more detail.

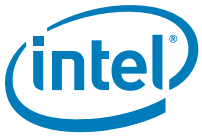


**Table 14-133.USB I/O Registers**

| BASE + Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00–01h | USBCMD | USB Command | 0000h | R/W |
| 02–03h | USBSTS | USB Status | 0020h | R/WC |
| 04–05h | USBINTR | USB Interrupt Enable | 0000h | R/W |
| 06–07h | FRNUM | Frame Number | 0000h | R/W (see Note 1) |
| 08–0Bh | FRBASEADD | Frame List Base Address | Undefined | R/W |
| 0Ch | SOFMOD | Start of Frame Modify | 40h | R/W |
| 0D–0Fh | — | **Reserved** | — | — |
| 10–11h | PORTSC0 | Port 0 Status/Control | 0080h | R/WC, RO, R/W (see Note 1) |
| 12–13h | PORTSC1 | Port 1 Status/Control | 0080h | R/WC, RO, R/W (see Note 1) |

**NOTES:**
1. These registers are WORD writable only. Byte writes to these registers have unpredictable effects.

## 14.2.1 USBCMD—USB Command Register

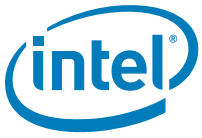
I/O Offset: Base + (00h–01h)Attribute:R/W

Default Value: 0000hSize:16 bits

The Command Register indicates the command to be executed by the serial bus host controller. Writing to the register causes a command to be executed. The table following the bit description provides additional information on the operation of the Run/Stop and Debug bits.

| Bit | Description |
|---|---|
| 15:7 | **Reserved** |
| 8 | **Loop Back Test Mode** — R/W.<br>0 = Disable loop back test mode.<br>1 = Chipset is in loop back test mode. When both ports are connected together, a write to one port will be seen on the other port and the data will be stored in I/O offset 18h. |
| 7 | **Max Packet (MAXP)** — R/W. This bit selects the maximum packet size that can be used for full speed bandwidth reclamation at the end of a frame. This value is used by the host controller to determine whether it should initiate another transaction based on the time remaining in the SOF counter. Use of reclamation packets larger than the programmed size will cause a Babble error if executed during the critical window at frame end. The Babble error results in the offending endpoint being stalled. Software is responsible for ensuring that any packet which could be executed under bandwidth reclamation be within this size limit.<br>0 = 32 bytes<br>1 = 64 bytes |

| Bit | Description |
|---|---|
| 6 | **Configure Flag (CF)** — R/W. This bit has no effect on the hardware. It is provided only as a semaphore service for software.<br>0 = Indicates that software has not completed host controller configuration.<br>1 = HCD software sets this bit as the last action in its process of configuring the host controller. |
| 5 | **Software Debug (SWDBG)** — R/W. The SWDBG bit must only be manipulated when the controller is in the stopped state. This can be determined by checking the HCHalted bit in the USBSTS register.<br>0 = Normal Mode.<br>1 = Debug mode. In SW Debug mode, the host controller clears the Run/Stop bit after the completion of each USB transaction. The next transaction is executed when software sets the Run/Stop bit back to 1. |
| 4 | **Force Global Resume (FGR)** — R/W.<br>0 = Software resets this bit to 0 after 20 ms has elapsed to stop sending the Global Resume signal. At that time all USB devices should be ready for bus activity. The 1 to 0 transition causes the port to send a low speed EOP signal. This bit will remain a 1 until the EOP has completed.<br>1 = Host controller sends the Global Resume signal on the USB, and sets this bit to 1 when a resume event (connect, disconnect, or K-state) is detected while in global suspend mode. |
| 3 | **Enter Global Suspend Mode (EGSM)** — R/W.<br>0 = Software resets this bit to 0 to come out of Global Suspend mode. Software writes this bit to 0 at the same time that Force Global Resume (bit 4) is written to 0 or after writing bit 4 to 0.<br>1 = Host controller enters the Global Suspend mode. No USB transactions occur during this time. The Host controller is able to receive resume signals from USB and interrupt the system. Software must ensure that the Run/Stop bit (bit 0) is cleared prior to setting this bit. |
| 2 | **Global Reset (GRESET)** — R/W.<br>0 = This bit is reset by the software after a minimum of 10 ms has elapsed as specified in Chapter 7 of the USB Specification.<br>1 = Global Reset. The host controller sends the global reset signal on the USB and then resets all its logic, including the internal hub registers. The hub registers are reset to their power on state. Chip Hardware Reset has the same effect as Global Reset (bit 2), except that the host controller does not send the Global Reset on USB. |
| 1 | **Host Controller Reset (HCRESET)** — R/W. The effects of HCRESET on Hub registers are slightly different from Chip Hardware Reset and Global USB Reset. The HCRESET affects bits [8,3:0] of the Port Status and Control Register (PORTSC) of each port. HCRESET resets the state machines of the host controller including the Connect/ Disconnect state machine (one for each port). When the Connect/Disconnect state machine is reset, the output that signals connect/disconnect are negated to 0, effectively signaling a disconnect, even if a device is attached to the port. This virtual disconnect causes the port to be disabled. This disconnect and disabling of the port causes bit 1 (connect status change) and bit 3 (port enable/disable change) of the PORTSC (D29:F0/F1/F2/F3:BASE + 10h) to get set. The disconnect also causes bit 8 of PORTSC to reset. About 64 bit times after HCRESET goes to 0, the connect and low-speed detect will take place, and bits 0 and 8 of the PORTSC will change accordingly.<br>0 = Reset by the host controller when the reset process is complete.<br>1 = Reset. When this bit is set, the host controller module resets its internal timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. |



| Bit | Description |
|---|---|
| 0 | **Run/Stop (RS)** — R/W. When set to 1, the Chipset proceeds with execution of the schedule. The Chipset continues execution as long as this bit is set. When this bit is cleared, the Chipset completes the current transaction on the USB and then halts. The HC Halted bit in the status register indicates when the host controller has finished the transaction and has entered the stopped state. The host controller clears this bit when the following fatal errors occur: consistency check failure, PCI Bus errors.<br>0 = Stop<br>1 = Run<br><br>**NOTE:** This bit should only be cleared if there are no active Transaction Descriptors in the executable schedule or software will reset the host controller prior to setting this bit again. |

**Table 14-134.Run/Stop, Debug Bit Interaction SWDBG (Bit 5), Run/Stop (Bit 0) Operation**

| SWDBG (Bit 5) | Run/Stop (Bit 0) | Description |
|---|---|---|
| 0 | 0 | If executing a command, the host controller completes the command and then stops. The 1.0 ms frame counter is reset and command list execution resumes from start of frame using the frame list pointer selected by the current value in the FRNUM register. (While Run/Stop=0, the FRNUM register (D29:F0/F1/F2/F3:BASE + 06h) can be reprogrammed). |
| 0 | 1 | Execution of the command list resumes from Start Of Frame using the frame list pointer selected by the current value in the FRNUM register. The host controller remains running until the Run/Stop bit is cleared (by software or hardware). |
| 1 | 0 | If executing a command, the host controller completes the command and then stops and the 1.0 ms frame counter is frozen at its current value. All status are preserved. The host controller begins execution of the command list from where it left off when the Run/Stop bit is set. |
| 1 | 1 | Execution of the command list resumes from where the previous execution stopped. The Run/Stop bit is set to 0 by the host controller when a TD is being fetched. This causes the host controller to stop again after the execution of the TD (single step). When the host controller has completed execution, the HC Halted bit in the Status Register is set. |

When the USB host controller is in Software Debug Mode (USBCMD Register bit 5=1), the single stepping software debug operation is as follows:

To Enter Software Debug Mode:

1. HCD puts host controller in Stop state by setting the Run/Stop bit to 0.
2. HCD puts host controller in Debug Mode by setting the SWDBG bit to 1.
3. HCD sets up the correct command list and Start Of Frame value for starting point in the Frame List Single Step Loop.
4. HCD sets Run/Stop bit to 1.

5. Host controller executes next active TD, sets Run/Stop bit to 0, and stops.
6. HCD reads the USBCMD register to check if the single step execution is completed (HCHalted=1).
7. HCD checks results of TD execution. Go to step 4 to execute next TD or step 8 to end Software Debug mode.
8. HCD ends Software Debug mode by setting SWDBG bit to 0.
9. HCD sets up normal command list and Frame List table.
10. HCD sets Run/Stop bit to 1 to resume normal schedule execution.

In Software Debug mode, when the Run/Stop bit is set, the host controller starts. When a valid TD is found, the Run/Stop bit is reset. When the TD is finished, the HCHalted bit in the USBSTS register (bit 5) is set.

The SW Debug mode skips over inactive TDs and only halts after an active TD has been executed. When the last active TD in a frame has been executed, the host controller waits until the next SOF is sent and then fetches the first TD of the next frame before halting.

This HCHalted bit can also be used outside of Software Debug mode to indicate when the host controller has detected the Run/Stop bit and has completed the current transaction. Outside of the Software Debug mode, setting the Run/Stop bit to 0 always resets the SOF counter so that when the Run/Stop bit is set the host controller starts over again from the frame list location pointed to by the Frame List Index (see FRNUM Register description) rather than continuing where it stopped.

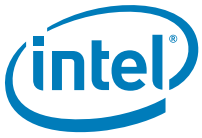
## 14.2.2 USBSTS—USB Status Register

I/O Offset: Base + (02h–03h)Attribute:R/WC

Default Value: 0020hSize:16 bits

This register indicates pending interrupts and various states of the host controller. The status resulting from a transaction on the serial bus is not indicated in this register.

| Bit | Description |
|---|---|
| 15:6 | **Reserved** |
| 5 | **HCHalted** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = The host controller has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the host controller hardware (debug mode or an internal error). Default. |



| Bit | Description |
|---|---|
| 4 | **Host Controller Process Error** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = The host controller has detected a fatal error. This indicates that the host controller suffered a consistency check failure while processing a Transfer Descriptor. An example of a consistency check failure would be finding an invalid PID field while processing the packet header portion of the TD. When this error occurs, the host controller clears the Run/Stop bit in the Command register (D29:F0/F1/F2/F3:BASE + 00h, bit 0) to prevent further schedule execution. A hardware interrupt is generated to the system. |
| 3 | **Host System Error** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = A serious error occurred during a host system access involving the host controller module. In a PCI system, conditions that set this bit to 1 include PCI Parity error, PCI Master Abort, and PCI Target Abort. When this error occurs, the host controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs. A hardware interrupt is generated to the system. |
| 2 | **Resume Detect (RSM_DET)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = The host controller received a “RESUME” signal from a USB device. This is only valid if the Host controller is in a global suspend state (Command register, D29:F0/F1/F2/F3:BASE + 00h, bit 3 = 1). |
| 1 | **USB Error Interrupt** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Completion of a USB transaction resulted in an error condition (e.g., error counter underflow). If the TD on which the error interrupt occurred also had its IOC bit (D29:F0/F1/F2/F3:BASE + 04h, bit 2) set, both this bit and Bit 0 are set. |
| 0 | **USB Interrupt (USBINT)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = The host controller sets this bit when the cause of an interrupt is a completion of a USB transaction whose Transfer Descriptor had its IOC bit set. Also set when a short packet is detected (actual length field in TD is less than maximum length field in TD), and short packet detection is enabled in that TD. |

### 14.2.3 USBINTR—USB Interrupt Enable Register

I/O Offset: Base + (04h–05h)Attribute:R/W

Default Value: 0000hSize:16 bits

This register enables and disables reporting of the corresponding interrupt to the software. When a bit is set and the corresponding interrupt is active, an interrupt is generated to the host. Fatal errors (host controller processor error, (D29:F0/F1/F2/F3:BASE + 02h, bit 4, USBSTS Register) cannot be disabled by the host controller. Interrupt sources that are disabled in this register still appear in the Status Register to allow the software to poll for events.

| Bit | Description |
|---|---|
| 15:5 | **Reserved** |
| 4 | **Scratchpad (SP)** — R/W. |
| 3 | **Short Packet Interrupt Enable** — R/W.<br>0 = Disabled.<br>1 = Enabled. |
| 2 | **Interrupt on Complete Enable (IOC)** — R/W.<br>0 = Disabled.<br>1 = Enabled. |
| 1 | **Resume Interrupt Enable** — R/W.<br>0 = Disabled.<br>1 = Enabled. |
| 0 | **Timeout/CRC Interrupt Enable** — R/W.<br>0 = Disabled.<br>1 = Enabled. |

### 14.2.4 FRNUM—Frame Number Register

I/O Offset: Base + (06–07h)Attribute:R/W (Writes must be Word Writes)

Default Value: 0000hSize:16 bits

Bits [10:0] of this register contain the current frame number that is included in the frame SOF packet. This register reflects the count value of the internal frame number counter. Bits [9:0] are used to select a particular entry in the Frame List during scheduled execution. This register is updated at the end of each frame time.

This register must be written as a word. Byte writes are not supported. This register cannot be written unless the host controller is in the STOPPED state as indicated by the HCHalted bit (D29:F0/F1/F2/F3:BASE + 02h, bit 5). A write to this register while the Run/Stop bit is set (D29:F0/F1/F2/F3:BASE + 00h, bit 0) is ignored.

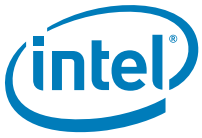
| Bit | Description |
|---|---|
| 15:11 | **Reserved** |
| 10:0 | **Frame List Current Index/Frame Number** — R/W. This field provides the frame number in the SOF Frame. The value in this register increments at the end of each time frame (approximately every 1 ms). In addition, bits [9:0] are used for the Frame List current index and correspond to memory address signals [11:2]. |

### 14.2.5 FRBASEADD—Frame List Base Address Register

I/O Offset: Base + (08h–0Bh)Attribute:R/W

Default Value: UndefinedSize:32 bits

This 32-bit register contains the beginning address of the Frame List in the system memory. HCD loads this register prior to starting the schedule execution by the host controller. When written, only the upper 20 bits are used. The lower 12 bits are written



as 0's (4-KB alignment). The contents of this register are combined with the frame number counter to enable the host controller to step through the Frame List in sequence. The two least significant bits are always 00. This requires DWord-alignment for all list entries. This configuration supports 1024 Frame List entries.

| Bit | Description |
|---|---|
| 31:12 | **Base Address** — R/W. These bits correspond to memory address signals [31:12], respectively. |
| 11:0 | **Reserved** |

## 14.2.6 SOFMOD—Start of Frame Modify Register

I/O Offset: Base + (0Ch)Attribute:R/W

Default Value: 40hSize:8 bits

This 1-byte register is used to modify the value used in the generation of SOF timing on the USB. Only the 7 least significant bits are used. When a new value is written into these 7 bits, the SOF timing of the next frame will be adjusted. This feature can be used to adjust out any offset from the clock source that generates the clock that drives the SOF counter. This register can also be used to maintain real time synchronization with the rest of the system so that all devices have the same sense of real time. Using this register, the frame length can be adjusted across the full range required by the USB specification. Its initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. It may be reprogrammed by USB system software at any time. Its value will take effect from the beginning of the next frame. This register is reset upon a host controller reset or global reset. Software must maintain a copy of its value for reprogramming if necessary.

| Bit | Description |
|---|---|
| 7 | **Reserved** |
| 6:0 | **SOF Timing Value** — R/W. Guidelines for the modification of frame time are contained in Chapter 7 of the USB Specification. The SOF cycle time (number of SOF counter clock periods to generate a SOF frame length) is equal to 11936 + value in this field. The default value is decimal 64 which gives a SOF cycle time of 12000. For a 12 MHz SOF counter clock input, this produces a 1 ms Frame period. The following table indicates what SOF Timing Value to program into this field for a certain frame period. |

| Frame Length (# 12 MHz Clocks) (decimal) | SOF Timing Value (this register) (decimal) |
|---|---|
| 11936 | 0 |
| 11937 | 1 |
| — | — |
| 11999 | 63 |
| 12000 | 64 |
| 12001 | 65 |
| — | — |
| 12062 | 126 |
| 12063 | 127 |

### 14.2.7 PORTSC[0,1]—Port Status and Control Register

I/O Offset: Port 0/2/4/6: Base + (10h–11h)Attribute:R/WC, RO,

Port 1/3/5/7: Base + (12h–13h)R/W (Word writes only)

Default Value: 0080hSize:16 bits

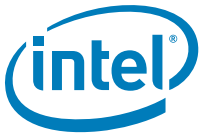
***Note:*** For Function 0, this applies to Chipset USB ports 0 and 1; for Function 1, this applies to Chipset USB ports 2 and 3; for Function 2, this applies to Chipset USB ports 4 and 5; and for Function 3, this applies to Chipset USB ports 6 and 7.

After a power-up reset, global reset, or host controller reset, the initial conditions of a port are: no device connected, Port disabled, and the bus line status is 00 (single-ended 0).

Port Reset and Enable Sequence

When software wishes to reset a USB device it will assert the Port Reset bit in the Port Status and Control register. The minimum reset signaling time is 10 mS and is enforced by software. To complete the reset sequence, software clears the port reset bit. The Intel UHCI controller must re-detect the port connect after reset signaling is complete before the controller will allow the port enable bit to de set by software. This time is approximately 5.3 uS. Software has several possible options to meet the timing requirement and a partial list is enumerated below:

- Iterate a short wait, setting the port enable bit and reading it back to see if the enable bit is set.
- Poll the connect status bit and wait for the hardware to recognize the connect prior to enabling the port.
- Wait longer than the hardware detect time after clearing the port reset and prior to enabling the port.



<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>15:13</td><td>Reserved — RO.</td></tr>
<tr><td>12</td><td>Suspend — R/W. This bit should not be written to a 1 if global suspend is active (bit 3=1 in the USBCMD register). Bit 2 and bit 12 of this register define the hub states as follows:<br><br>
<table>
<tr><th>Bits [12,2]</th><th>Hub State</th></tr>
<tr><td>X,0</td><td>Disable</td></tr>
<tr><td>0, 1</td><td>Enable</td></tr>
<tr><td>1, 1</td><td>Suspend</td></tr>
</table>
When in suspend state, downstream propagation of data is blocked on this port, except for single-ended 0 resets (global reset and port reset). The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB.<br>
1 = Port in suspend state.<br>
0 = Port not in suspend state.<br><br>
NOTE: Normally, if a transaction is in progress when this bit is set, the port will be suspended when the current transaction completes. However, in the case of a</td></tr>
<tr><td>11</td><td>Overcurrent Indicator — R/WC. Set by hardware.<br>
0 = Software clears this bit by writing a 1 to it.<br>
1 = Overcurrent pin has gone from inactive to active on this port.</td></tr>
<tr><td>10</td><td>Overcurrent Active — RO. This bit is set and cleared by hardware.<br>
0 = Indicates that the overcurrent pin is inactive (high).<br>
1 = Indicates that the overcurrent pin is active (low).</td></tr>
<tr><td>9</td><td>Port Reset — R/W.<br>
0 = Port is not in Reset.<br>
1 = Port is in Reset. When set, the port is disabled and sends the USB Reset signaling.</td></tr>
<tr><td>8</td><td>Low Speed Device Attached (LS) — RO.<br>
0 = Full speed device is attached.<br>
1 = Low speed device is attached to this port.</td></tr>
<tr><td>7</td><td>Reserved — RO. Always read as 1.</td></tr>
<tr><td>6</td><td>Resume Detect (RSM_DET) — R/W. Software sets this bit to a 1 to drive resume signaling. The host controller sets this bit to a 1 if a J-to-K transition is detected for at least 32 microseconds while the port is in the Suspend state. The Chipset will then reflect the K-state back onto the bus as long as the bit remains a 1, and the port is still in the suspend state (bit 12,2 are ‘11’). Writing a 0 (from 1) causes the port to send a low speed EOP. This bit will remain a 1 until the EOP has completed.<br>
0 = No resume (K-state) detected/driven on port.<br>
1 = Resume detected/driven on port.</td></tr>
<tr><td>5:4</td><td>Line Status — RO. These bits reflect the D+ (bit 4) and D– (bit 5) signals lines’ logical levels. These bits are used for fault detect and recovery as well as for USB diagnostics. This field is updated at EOF2 time (See Chapter 11 of the USB Specification).</td></tr>
</table>

| Bit | Description |
|---|---|
| 3 | **Port Enable / Disable Change** — R/WC. For the root hub, this bit gets set only when a port is disabled due to disconnect on that port or due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification).<br>0 = No change. Software clears this bit by writing a 1 to the bit location.<br>1 = Port enabled/disabled status has changed. |
| 2 | **Port Enabled / Disabled (PORT_EN)** — R/W. Ports can be enabled by host software only. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes and that there may be a delay in disabling or enabling a port if there is a transaction currently in progress on the USB.<br>0 = Disable<br>1 = Enable |
| 1 | **Connect Status Change** — R/WC. This bit indicates that a change has occurred in the port's Current Connect Status (see bit 0). The hub device sets this bit for any changes to the port device connect status, even if system software has not cleared a connect status change. If, for example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be setting" an already-set bit (i.e., the bit will remain set). However, the hub transfers the change bit only once when the host controller requests a data transfer to the Status Change endpoint. System software is responsible for determining state change history in such a case.<br>0 = No change. Software clears this bit by writing a 1 to it.<br>1 = Change in Current Connect Status. |
| 0 | **Current Connect Status** — RO. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set.<br>0 = No device is present.<br>1 = Device is present on port. |

§

# 15 SATA Controller Registers (D31:F2)

## 15.1 PCI Configuration Registers (SATA–D31:F2)

***Note:*** Address locations that are not shown should be treated as **Reserved**.

All of the SATA registers are in the core well. None of the registers can be locked.

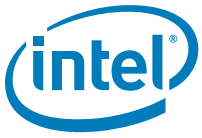
**Table 15-135.SATA Controller PCI Register Address Map (SATA–D31:F2) (Sheet 1 of 2)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00h–01h | VID | Vendor Identification | 8086h | RO |
| 02h–03h | DID | Device Identification | See register description. | RO |
| 04h–05h | PCICMD | PCI Command | 0000h | R/W, RO |
| 06h–07h | PCISTS | PCI Status | 02B0h | R/WC, RO |
| 08h | RID | Revision Identification | See register description. | RO |
| 09h | PI | Programming Interface | See register description. | See register description |
| 0Ah | SCC | Sub Class Code | See register description | See register description |
| 0Bh | BCC | Base Class Code | 01h | RO |
| 0Dh | PMLT | Primary Master Latency Timer | 00h | RO |
| 10h–13h | PCMD_BAR | Primary Command Block Base Address | 00000001h | R/W, RO |
| 14h–17h | PCNL_BAR | Primary Control Block Base Address | 00000001h | R/W, RO |
| 18h–1Bh | SCMD_BAR | Secondary Command Block Base Address | 00000001h | R/W, RO |
| 1Ch–1Fh | SCNL_BAR | Secondary Control Block Base Address | 00000001h | R/W, RO |
| 20h–23h | BAR | Legacy Bus Master Base Address | 00000001h | R/W, RO |
| 24h–27h | ABAR | AHCI Base Address | 00000000h | See register description |
| 2Ch–2Dh | SVID | Subsystem Vendor Identification | 0000h | R/WO |
| 2Eh–2Fh | SID | Subsystem Identification | 0000h | R/WO |
| 34h | CAP | Capabilities Pointer | 80h | RO |
| 3Ch | INT_LN | Interrupt Line | 00h | R/W |
| 3Dh | INT_PN | Interrupt Pin | See register description. | RO |

### Table 15-135.SATA Controller PCI Register Address Map (SATA–D31:F2) (Sheet 2 of 2)

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 40h–41h | IDE_TIMP | Primary IDE Timing | 0000h | R/W |
| 42h–43h | IDE_TIMS | Secondary IDE Timing | 0000h | R/W |
| 44h | SIDETIM | Slave IDE Timing | 00h | R/W |
| 48h | SDMA_CNT | Synchronous DMA Control | 00h | R/W |
| 4Ah–4Bh | SDMA_TIM | Synchronous DMA Timing | 0000h | R/W |
| 54h–57h | IDE_CONFIG | IDE I/O Configuration | 00000000h | R/W |
| 70h–71h | PID | PCI Power Management Capability ID | See register description | RO |
| 72h–73h | PC | PCI Power Management Capabilities | 4002h | RO |
| 74h–75h | PMCS | PCI Power Management Control and Status | 0000h | R/W, RO, R/WC |
| 80h–81h | MSICI | Message Signaled Interrupt Capability ID | 7005h | RO |
| 82h–83h | MSIMC | Message Signaled Interrupt Message Control | 0000h | RO, R/W |
| 84h–87h | MSIMA | Message Signaled Interrupt Message Address | 00000000h | RO, R/W |
| 88h–89h | MSIMD | Message Signaled Interrupt Message Data | 0000h | R/W |
| 90h | MAP | Address Map | 00h | R/W |
| 92h–93h | PCS | Port Control and Status | 0000h | R/W, RO, R/WC |
| 94h–97h | SIR | SATA Initialization Register | 00000000h | R/W |
| A0h | SIRI | SATA Indexed Registers Index | 00h | R/W |
| A4h | STRD | SATA Indexed Register Data | XXXXXXXXh | R/W |
| A8h–ABh | SCAP0 | SATA Capability Register 0 | 00100012h | RO |
| ACh–AFh | SCAP1 | SATA Capability Register 1 | 00000048h | RO |
| C0h | ATC | APM Trapping Control | 00h | R/W |
| C4h | ATS | ATM Trapping Status | 00h | R/WC |
| D0h–D3h | SP | Scratch Pad | 00000000h | R/W |
| E0h–E3h | BFCS | BIST FIS Control/Status | 00000000h | R/W, R/WC |
| E4h–E7h | BFTD1 | BIST FIS Transmit Data, DW1 | 00000000h | R/W |
| E8h–EBh | BFTD2 | BIST FIS Transmit Data, DW2 | 00000000h | R/W |

**NOTE:** The Chipset SATA controller is not arbitrated as a PCI device; therefore, it does not need a master latency timer.



### 15.1.1 VID—Vendor Identification Register (SATA—D31:F2)

| | | | |
|---|---|---|---|
| Offset Address: | 00h–01h | Attribute: | RO |
| Default Value: | 8086h | Size: | 16 bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:0 | **Vendor ID** — RO. This is a 16-bit value assigned to Intel. Intel VID = 8086h |

### 15.1.2 DID—Device Identification Register (SATA—D31:F2)

| | | | |
|---|---|---|---|
| Offset Address: | 02h–03h | Attribute: | RO |
| Default Value: | See bit description | Size: | 16 bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:0 | **Device ID** — RO. This is a 16-bit value assigned to the Chipset SATA controller.<br>**NOTE:** The value of this field will change dependent upon the value of the MAP Register. |

### 15.1.3 PCICMD—PCI Command Register (SATA–D31:F2)

| | | | |
|---|---|---|---|
| Address Offset: | 04h–05h | Attribute: | RO, R/W |
| Default Value: | 0000h | Size: | 16 bits |

| Bit | Description |
|---|---|
| 15:11 | **Reserved** |
| 10 | **Interrupt Disable** — R/W. This disables pin-based INTx# interrupts. This bit has no effect on MSI operation.<br>0 = Internal INTx# messages are generated if there is an interrupt and MSI is not enabled.<br>1 = Internal INTx# messages will not be generated. |
| 9 | **Fast Back to Back Enable (FBE)** — RO. **Reserved** as 0. |
| 8 | SERR# Enable **(SERR_EN)** — RO. **Reserved** as 0. |
| 7 | **Wait Cycle Control (WCC)** — RO. **Reserved** as 0. |
| 6 | **Parity Error Response (PER)** — R/W.<br>0 = Disabled. SATA controller will not generate PERR# when a data parity error is detected.<br>1 = Enabled. SATA controller will generate PERR# when a data parity error is detected. |
| 5 | **VGA Palette Snoop (VPS)** — RO. **Reserved** as 0. |
| 4 | **Postable Memory Write Enable (PMWE)** — RO. **Reserved** as 0. |
| 3 | **Special Cycle Enable (SCE)** — RO. **Reserved** as 0. |

| Bit | Description |
|---|---|
| 2 | **Bus Master Enable (BME)** — R/W. This bit controls the chipset's ability to act as a PCI master for IDE Bus Master transfers. This bit does not impact the generation of completions for split transaction commands. |
| 1 | **Memory Space Enable (MSE)** — R/W / RO. This bit controls access to the SATA controller's target memory space (for AHCI). |
| 0 | **I/O Space Enable (IOSE)** — R/W. This bit controls access to the I/O space registers.<br>0 = Disables access to the Legacy or Native IDE ports (both Primary and Secondary) as well as the Bus Master I/O registers.<br>1 = Enable. Note that the Base Address register for the Bus Master registers should be programmed before this bit is set. |

### 15.1.4 PCISTS — PCI Status Register (SATA–D31:F2)

Address Offset: 06h–07h
Default Value: 02B0h
Attribute: R/WC, RO
Size: 16 bits

***Note:*** For the writable bits, software must write a 1 to clear bits that are set. Writing a 0 to the bit has no effect.

| Bit | Description |
|---|---|
| 15 | **Detected Parity Error (DPE)** — R/WC.<br>0 = No parity error detected by SATA controller.<br>1 = SATA controller detects a parity error on its interface. |
| 14 | **Signaled System Error (SSE)** — RO. **Reserved** as 0. |
| 13 | **Received Master Abort (RMA)** — R/WC.<br>0 = Master abort Not generated.<br>1 = SATA controller, as a master, generated a master abort. |
| 12 | **Reserved** as 0 — RO. |
| 11 | **Signaled Target Abort (STA)** — RO. **Reserved** as 0. |
| 10:9 | DEVSEL# Timing Status (DEV_STS) — RO.<br>01 = Hardwired; Controls the device select time for the SATA controller's PCI interface. |
| 8 | **Data Parity Error Detected (DPED)** — RO. For Chipset, this bit can only be set on read completions received from SiBUS where there is a parity error.<br>0 = Data parity error **Not** detected.<br>1 = SATA controller, as a master, either detects a parity error or sees the parity error line asserted, and the parity error response bit (bit 6 of the command register) is set. |
| 7 | **Fast Back to Back Capable (FB2BC)** — RO. **Reserved** as 1. |
| 6 | **User Definable Features (UDF)** — RO. **Reserved** as 0. |
| 5 | **66MHz Capable (66MHZ_CAP)** — RO. **Reserved** as 1. |

| Bit | Description |
|---|---|
| 4 | **Capabilities List (CAP_LIST)** — RO. This bit indicates the presence of a capabilities list. The minimum requirement for the capabilities list must be PCI power management for the SATA controller. |
| 3 | **Interrupt Status (INTS)** — RO. Reflects the state of INTx# messages.<br>0 = Interrupt is cleared (independent of the state of Interrupt Disable bit in the command register [offset 04h]).<br>1 = Interrupt is to be asserted |
| 2:0 | **Reserved** |

## 15.1.5 RID—Revision Identification Register (SATA—D31:F2)

Offset Address: 08h Attribute: RO
Default Value: See bit description Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Revision ID — RO. |

## 15.1.6 PI—Programming Interface Register (SATA–D31:F2)

### 15.1.6.1 When Sub Class Code Register (D31:F2:Offset 0Ah) = 01h

Address Offset: 09h Attribute: R/W, RO
Default Value: See bit description Size: 8 bits

| Bit | Description |
|---|---|
| 7 | This read-only bit is a 1 to indicate that the Chipset supports bus master operation |
| 6:4 | **Reserved**. Will always return 0. |
| 3 | **Secondary Mode Native Capable (SNC)** — RO.<br>0 = Secondary controller only supports legacy mode.<br>1 = Secondary controller supports both legacy and native modes.<br>The MAP.MV (D31:F2:Offset 90:bits 1:0) must be program as 00b, and this bit is reports as a 1. |
| 2 | **Secondary Mode Native Enable (SNE)** — R/W / RO.<br>Determines the mode that the secondary channel is operating in.<br>0 = Secondary controller operating in legacy (compatibility) mode<br>1 = Secondary controller operating in native PCI mode.<br>When MAP.MV (D31:F2:Offset 90:bits 1:0) is any value other than 00b, this bit is read-only (RO). Software is responsible for clearing this bit before entering combined mode. The MAP.MV must be program as 00b, and this bit is read/write (R/W).<br>If this bit is set by software, then the PNE bit (bit 0 of this register) must also be set by software. While in theory these bits can be programmed separately, such a configuration is not supported by hardware. |
| 1 | **Primary Mode Native Capable (PNC)** — RO.<br>0 = Primary controller only supports legacy mode.<br>1 = Primary controller supports both legacy and native modes.<br>The MAP.MV (D31:F2:Offset 90:bits 1:0) must be program as 00b, and this bit is reports as a 1. |

| Bit | Description |
|---|---|
| 0 | **Primary Mode Native Enable (PNE)** — R/W / RO.<br>Determines the mode that the primary channel is operating in.<br>0 = Primary controller operating in legacy (compatibility) mode.<br>1 = Primary controller operating in native PCI mode.<br>When MAP.MV (D31:F2:Offset 90:bits 1:0) is any value other than 00b, this bit is read-only (RO). Software is responsible for clearing this bit before entering combined mode. The MAP.MV must be program as 00b, and this bit is read/write (R/W).<br>If this bit is set by software, then the SNE bit (bit 2 of this register) must also be set by software. While in theory these bits can be programmed separately, such a configuration is not supported by hardware. |

#### 15.1.6.2 When Sub Class Code Register (D31:F2:Offset 0Ah) = 06h

Address Offset: 09h Attribute: RO
Default Value: 01h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Interface (IF)** — RO.<br>Indicates the SATA Controller supports AHCI, rev 1.0. |

### 15.1.7 SCC—Sub Class Code Register (SATA–D31:F2)

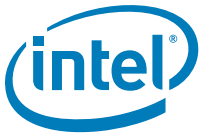
Address Offset: 0Ah Attribute: RO
Default Value: See bit description Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Sub Class Code (SCC)**<br>This field specifies the sub-class code of the controller, per the table below:<br>**MAP.SMS (D31:F2:Offset 90h:bit 7:6)** / **SCC Register Value**<br>00b / 01h (IDE Controller)<br>01b / 06h (AHCI Controller) |

### 15.1.8 BCC—Base Class Code Register (SATA–D31:F2SATA–D31:F2)

Address Offset: 0Bh Attribute: RO
Default Value: 01h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Base Class Code (BCC)** — RO.<br>01h = Mass storage device |



### 15.1.9 PMLT—Primary Master Latency Timer Register (SATA–D31:F2)

Address Offset: 0Dh Attribute: RO
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Master Latency Timer Count (MLTC)** — RO.<br>00h = Hardwired. The SATA controller is implemented internally, and is not arbitrated as a PCI device, so it does not need a Master Latency Timer. |

### 15.1.10 PCMD_BAR—Primary Command Block Base Address Register (SATA–D31:F2)

Address Offset: 10h–13h Attribute: R/W, RO
Default Value: 00000001h Size: 32 bits

| Bit | Description |
|---|---|
| 31:16 | **Reserved** |
| 15:3 | **Base Address** — R/W. This field provides the base address of the I/O space (8 consecutive I/O locations). |
| 2:1 | **Reserved** |
| 0 | Resource Type Indicator (RTE) — RO. Hardwired to 1 to indicate a request for I/O space. |

**NOTE:** This 8-byte I/O space is used in native mode for the Primary Controller's Command Block.

### 15.1.11 PCNL_BAR—Primary Control Block Base Address Register (SATA–D31:F2)

Address Offset: 14h–17h Attribute: R/W, RO
Default Value: 00000001h Size: 32 bits

| Bit | Description |
|---|---|
| 31:16 | **Reserved** |
| 15:2 | **Base Address** — R/W. This field provides the base address of the I/O space (4 consecutive I/O locations). |
| 1 | **Reserved** |
| 0 | **Resource Type Indicator (RTE)** — RO. Hardwired to 1 to indicate a request for I/O space. |

**NOTE:** This 4-byte I/O space is used in native mode for the Primary Controller's Command Block.

### 15.1.12 SCMD_BAR—Secondary Command Block Base Address Register (IDE D31:F1)

Address Offset: 18h–1Bh
Default Value: 00000001h
Attribute: R/W, RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:16 | **Reserved** |
| 15:3 | **Base Address** — R/W. This field provides the base address of the I/O space (8 consecutive I/O locations). |
| 2:1 | **Reserved** |
| 0 | **Resource Type Indicator (RTE)** — RO. Hardwired to 1 to indicate a request for I/O space. |

**NOTE:** This 4-byte I/O space is used in native mode for the Secondary Controller's Command Block.

### 15.1.13 SCNL_BAR—Secondary Control Block Base Address Register (IDE D31:F1)

Address Offset: 1Ch–1Fh
Default Value: 00000001h
Attribute: R/W, RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:16 | **Reserved** |
| 15:2 | **Base Address** — R/W. This field provides the base address of the I/O space (4 consecutive I/O locations). |
| 1 | **Reserved** |
| 0 | **Resource Type Indicator (RTE)** — RO. Hardwired to 1 to indicate a request for I/O space. |

**NOTE:** This 4-byte I/O space is used in native mode for the Secondary Controller's Command Block.

### 15.1.14 BAR — Legacy Bus Master Base Address Register (SATA–D31:F2)

Address Offset: 20h–23h
Default Value: 00000001h
Attribute: R/W, RO
Size: 32 bits

The Bus Master IDE interface function uses Base Address register 5 to request a 16-byte IO space to provide a software interface to the Bus Master functions. Only 12 bytes are actually used (6 bytes for primary, 6 bytes for secondary). Only bits [15:4] are used to decode the address.

| Bit | Description |
|---|---|
| 31:16 | **Reserved** |
| 15:4 | **Base Address** — R/W. This field provides the base address of the I/O space (16 consecutive I/O locations). |
| 3:1 | **Reserved** |

| Bit | Description |
|---|---|
| 0 | **Resource Type Indicator (RTE)** — RO. Hardwired to 1 to indicate a request for I/O space. |

## 15.1.15 ABAR — AHCI Base Address Register (SATA–D31:F2)

### 15.1.15.1 Non AHCI Capable (Chipset Feature Supported Components Only)

Address Offset: 24h–27h Attribute: RO
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Reserved** |

### 15.1.15.2 AHCI Capable

Address Offset: 24h–27h Attribute: R/W, RO
Default Value: 00000000h Size: 32 bits

This register allocates space for the memory registers defined in Section 15.3. For non-ACHI capable Chipset components, this register is reserved and read only, unless the SCRAE bit (offset 94h:bit 9) is set, in which case the register follows the definition given in Section 15.1.15.2.

| Bit | Description |
|---|---|
| 31:10 | **Base Address (BA)** — R/W. Base address of register memory space (aligned to 1 KB) |
| 9:4 | **Reserved** |
| 3 | **Prefetchable (PF)** — RO. Indicates that this range is not pre-fetchable |
| 2:1 | Type (TP) — RO. Indicates that this range can be mapped anywhere in 32-bit address space. |
| 0 | **Resource Type Indicator (RTE)** — RO. Hardwired to 0 to indicate a request for register memory space. |

**NOTES:**
1. When the MAP.MV register is programmed for combined mode (00b), this register is RO. Software is responsible for clearing this bit before entering combined mode.
2. The ABAR register must be set to a value of 0001_0000h or greater.

## 15.1.16 SVID—Subsystem Vendor Identification Register (SATA–D31:F2)

Address Offset: 2Ch–2Dh Attribute: R/WO
Default Value: 0000h Size: 16 bits
Lockable: No Power Well: Core

| Bit | Description |
|---|---|
| 15:0 | **Subsystem Vendor ID (SVID)** — R/WO. Value is written by BIOS. No hardware action taken on this value. |

### 15.1.17 SID—Subsystem Identification Register (SATA–D31:F2)

| | | | |
|---|---|---|---|
| Address Offset: | 2Eh–2Fh | Attribute: | R/WO |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:0 | **Subsystem ID (SID)** — R/WO. Value is written by BIOS. No hardware action taken on this value. |

### 15.1.18 CAP—Capabilities Pointer Register (SATA–D31:F2)

| | | | |
|---|---|---|---|
| Address Offset: | 34h | Attribute: | RO |
| Default Value: | 80h | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:0 | **Capabilities Pointer (CAP_PTR)** — RO. Indicates that the first capability pointer offset is 80h. This value changes to 70h if the MAP.MV register (Dev 31:F2:90h, bits 1:0) in configuration space indicates that the SATA function and PATA functions are combined (values of 10b or 10b) or Sub Class Code (CC.SCC) (Dev 31:F2:0Ah) is configure as IDE mode (value of 01). |

### 15.1.19 INT_LN—Interrupt Line Register (SATA–D31:F2)

| | | | |
|---|---|---|---|
| Address Offset: | 3Ch | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:0 | **Interrupt Line** — R/W. This field is used to communicate to software the interrupt line that the interrupt pin is connected to. |

### 15.1.20 INT_PN—Interrupt Pin Register (SATA–D31:F2)

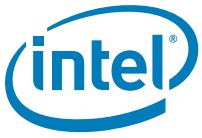
| | | | |
|---|---|---|---|
| Address Offset: | 3Dh | Attribute: | RO |
| Default Value: | See Register Description | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:0 | **Interrupt Pin** — RO. This reflects the value of D31IP.SIP (Chipset Config Registers:Offset 3100h: bits 11:8). |

### 15.1.21 IDE_TIMP — Primary IDE Timing Register (SATA–D31:F2)

| | | | |
|---|---|---|---|
| Address Offset: | Primary: 40h–41h<br>Secondary: 42h–43h | Attribute: | R/W |
| Default Value: | 0000h | Size: | 16 bits |

This register controls the timings driven on the IDE cable for PIO and 8237 style DMA transfers. It also controls operation of the buffer for PIO transfers.



***Note:*** This register is R/W to maintain software compatibility and enable parallel ATA functionality when the PCI functions are combined. These bits have no effect on SATA operation unless otherwise noted.

| Bit | Description |
|---|---|
| 15 | **IDE Decode Enable (IDE)** — R/W. Individually enable/disable the Primary or Secondary decode.<br>0 = Disable.<br>1 = Enables the Chipset to decode the associated Command Blocks (1F0–1F7h for primary, 170–177h for secondary) and Control Block (3F6h for primary and 376h for secondary).<br>This bit effects the IDE decode ranges for both legacy and native-Mode decoding.<br><br>**NOTE:** This bit affects SATA operation in both combined and non-combined ATA modes. See Section 5.17 - Volume 1 for more on ATA modes of operation. |
| 14 | **Drive 1 Timing Register Enable (SITRE)** — R/W.<br>0 = Use bits 13:12, 9:8 for both drive 0 and drive 1.<br>1 = Use bits 13:12, 9:8 for drive 0, and use the Slave IDE Timing register for drive 1 |
| 13:12 | **IORDY Sample Point (ISP)** — R/W. The setting of these bits determines the number of PCI clocks between IDE IOR#/IOW# assertion and the first IORDY sample point.<br>00 = 5 clocks<br>01 = 4 clocks<br>10 = 3 clocks<br>11 = **Reserved** |
| 11:10 | **Reserved** |
| 9:8 | **Recovery Time (RCT)** — R/W. The setting of these bits determines the minimum number of PCI clocks between the last IORDY sample point and the IOR#/IOW# strobe of the next cycle.<br>00 = 4 clocks<br>01 = 3 clocks<br>10 = 2 clocks<br>11 = 1 clock |
| 7 | **Drive 1 DMA Timing Enable (DTE1)** — R/W.<br>0 = Disable.<br>1 = Enable the fast timing mode for DMA transfers only for this drive. PIO transfers to the IDE data port will run in compatible timing. |
| 6 | **Drive 1 Prefetch/Posting Enable (PPE1)** — R/W.<br>0 = Disable.<br>1 = Enable Prefetch and posting to the IDE data port for this drive. |
| 5 | **Drive 1 IORDY Sample Point Enable (IE1)** — R/W.<br>0 = Disable IORDY sampling for this drive.<br>1 = Enable IORDY sampling for this drive. |
| 4 | **Drive 1 Fast Timing Bank (TIME1)** — R/W.<br>0 = Accesses to the data port will use compatible timings for this drive.<br>1 = When this bit =1 and bit 14 = 0, accesses to the data port will use bits 13:12 for the IORDY sample point, and bits 9:8 for the recovery time. When this bit = 1 and bit 14 = 1, accesses to the data port will use the IORDY sample point and recover time specified in the slave IDE timing register. |
| 3 | **Drive 0 DMA Timing Enable (DTE0)** — R/W.<br>0 = Disable<br>1 = Enable fast timing mode for DMA transfers only for this drive. PIO transfers to the IDE data port will run in compatible timing. |

| Bit | Description |
|---|---|
| 2 | **Drive 0 Prefetch/Posting Enable (PPE0)** — R/W.<br>0 = Disable prefetch and posting to the IDE data port for this drive.<br>1 = Enable prefetch and posting to the IDE data port for this drive. |
| 1 | **Drive 0 IORDY Sample Point Enable (IE0)** — R/W.<br>0 = Disable IORDY sampling is disabled for this drive.<br>1 = Enable IORDY sampling for this drive. |
| 0 | **Drive 0 Fast Timing Bank (TIME0)** — R/W.<br>0 = Accesses to the data port will use compatible timings for this drive.<br>1 = Accesses to the data port will use bits 13:12 for the IORDY sample point, and bits 9:8 for the recovery time |

## 15.1.22 IDE_TIMS — Slave IDE Timing Register (SATA–D31:F2)

Address Offset: 44h
Default Value: 00h
Attribute: R/W
Size: 8 bits

***Note:*** This register is R/W to maintain software compatibility and enable parallel ATA functionality when the PCI functions are combined. These bits have no effect on SATA operation unless otherwise noted.

| Bit | Description |
|---|---|
| 7:6 | **Secondary Drive 1 IORDY Sample Point (SISP1)** — R/W. This field determines the number of PCI clocks between IDE IOR#/IOW# assertion and the first IORDY sample point, if the access is to drive 1 data port and bit 14 of the IDE timing register for secondary is set.<br>00 = 5 clocks<br>01 = 4 clocks<br>10 = 3 clocks<br>11 = **Reserved** |
| 5:4 | **Secondary Drive 1 Recovery Time (SRCT1)** — R/W. This field determines the minimum number of PCI clocks between the last IORDY sample point and the IOR#/IOW# strobe of the next cycle, if the access is to drive 1 data port and bit 14 of the IDE timing register for secondary is set.<br>00 = 4 clocks<br>01 = 3 clocks<br>10 = 2 clocks<br>11 = 1 clocks |
| 3:2 | **Primary Drive 1 IORDY Sample Point (PISP1)** — R/W. This field determines the number of PCI clocks between IOR#/IOW# assertion and the first IORDY sample point, if the access is to drive 1 data port and bit 14 of the IDE timing register for primary is set.<br>00 = 5 clocks<br>01 = 4 clocks<br>10 = 3 clocks<br>11 = **Reserved** |

| Bit | Description |
|---|---|
| 1:0 | **Primary Drive 1 Recovery Time (PRCT1)** — R/W. This field determines the minimum number of PCI clocks between the last IORDY sample point and the IOR#/ IOW# strobe of the next cycle, if the access is to drive 1 data port and bit 14 of the IDE timing register for primary is set.<br>00 = 4 clocks<br>01 = 3 clocks<br>10 = 2 clocks<br>11 = 1 clocks |

## 15.1.23 SDMA_CNT—Synchronous DMA Control Register (SATA–D31:F2)

Address Offset: 48h
Default Value: 00h
Attribute: R/W
Size: 8 bits

***Note:*** This register is R/W to maintain software compatibility and enable parallel ATA functionality when the PCI functions are combined. These bits have no effect on SATA operation unless otherwise noted.

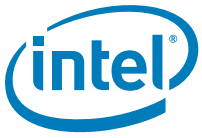
| Bit | Description |
|---|---|
| 7:4 | **Reserved** |
| 3 | **Secondary Drive 1 Synchronous DMA Mode Enable (SSDE1)** — R/W.<br>0 = Disable (default)<br>1 = Enable Synchronous DMA mode for secondary channel drive 1 |
| 2 | **Secondary Drive 0 Synchronous DMA Mode Enable (SSDE0)** — R/W.<br>0 = Disable (default)<br>1 = Enable Synchronous DMA mode for secondary drive 0. |
| 1 | **Primary Drive 1 Synchronous DMA Mode Enable (PSDE1)** — R/W.<br>0 = Disable (default)<br>1 = Enable Synchronous DMA mode for primary channel drive 1 |
| 0 | **Primary Drive 0 Synchronous DMA Mode Enable (PSDE0)** — R/W.<br>0 = Disable (default)<br>1 = Enable Synchronous DMA mode for primary channel drive 0 |

## 15.1.24 SDMA_TIM—Synchronous DMA Timing Register (SATA–D31:F2)

Address Offset: 4Ah–4Bh
Default Value: 0000h
Attribute: R/W
Size: 16 bits

***Note:*** This register is R/W to maintain software compatibility and enable parallel ATA functionality when the PCI functions are combined. These bits have no effect on SATA operation, unless otherwise noted.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>15:14</td><td>Reserved</td></tr>
<tr><td>13:12</td><td>Secondary Drive 1 Cycle Time (SCT1) — R/W. For Ultra ATA mode. The setting of these bits determines the minimum write strobe cycle time (CT). The DMARDY#-to-STOP (RP) time is also determined by the setting of these bits.
<table>
<tr><th>SCB1 = 0<br>(33 MHz clk)</th><th>SCB1 = 1<br>(66 MHz clk)</th><th>FAST_SCB1 = 1<br>(133 MHz clk)</th></tr>
<tr><td>00 = CT 4 clocks, RP 6 clocks</td><td>00 = Reserved</td><td>00 = Reserved</td></tr>
<tr><td>01 = CT 3 clocks, RP 5 clocks</td><td>01 = CT 3 clocks, RP 8 clocks</td><td>01 = CT 3 clocks, RP 16 clocks</td></tr>
<tr><td>10 = CT 2 clocks, RP 4 clocks</td><td>10 = CT 2 clocks, RP 8 clocks</td><td>10 = Reserved</td></tr>
<tr><td>11 = Reserved</td><td>11 = Reserved</td><td>11 = Reserved</td></tr>
</table></td></tr>
<tr><td>11:10</td><td>Reserved</td></tr>
<tr><td>9:8</td><td>Secondary Drive 0 Cycle Time (SCT0) — R/W. For Ultra ATA mode. The setting of these bits determines the minimum write strobe cycle time (CT). The DMARDY#-to-STOP (RP) time is also determined by the setting of these bits.
<table>
<tr><th>SCB1 = 0<br>(33 MHz clk)</th><th>SCB1 = 1<br>(66 MHz clk)</th><th>FAST_SCB1 = 1<br>(133 MHz clk)</th></tr>
<tr><td>00 = CT 4 clocks, RP 6 clocks</td><td>00 = Reserved</td><td>00 = Reserved</td></tr>
<tr><td>01 = CT 3 clocks, RP 5 clocks</td><td>01 = CT 3 clocks, RP 8 clocks</td><td>01 = CT 3 clocks, RP 16 clocks</td></tr>
<tr><td>10 = CT 2 clocks, RP 4 clocks</td><td>10 = CT 2 clocks, RP 8 clocks</td><td>10 = Reserved</td></tr>
<tr><td>11 = Reserved</td><td>11 = Reserved</td><td>11 = Reserved</td></tr>
</table></td></tr>
<tr><td>7:6</td><td>Reserved</td></tr>
<tr><td>5:4</td><td>Primary Drive 1 Cycle Time (PCT1) — R/W. For Ultra ATA mode, the setting of these bits determines the minimum write strobe cycle time (CT). The DMARDY#-to-STOP (RP) time is also determined by the setting of these bits.
<table>
<tr><th>PCB1 = 0<br>(33 MHz clk)</th><th>PCB1 = 1<br>(66 MHz clk)</th><th>FAST_PCB1 = 1 (133 MHz clk)</th></tr>
<tr><td>00 = CT 4 clocks, RP 6 clocks</td><td>00 = Reserved</td><td>00 = Reserved</td></tr>
<tr><td>01 = CT 3 clocks, RP 5 clocks</td><td>01 = CT 3 clocks, RP 8 clocks</td><td>01 = CT 3 clocks, RP 16 clocks</td></tr>
<tr><td>10 = CT 2 clocks, RP 4 clocks</td><td>10 = CT 2 clocks, RP 8 clocks</td><td>10 = Reserved</td></tr>
<tr><td>11 = Reserved</td><td>11 = Reserved</td><td>11 = Reserved</td></tr>
</table></td></tr>
<tr><td>3:2</td><td>Reserved</td></tr>
</table>



| Bit | Description |
|---|---|
| 1:0 | **Primary Drive 0 Cycle Time (PCT0)** — R/W. For Ultra ATA mode, the setting of these bits determines the minimum write strobe cycle time (CT). The DMARDY#-to-STOP (RP) time is also determined by the setting of these bits. |

| PCB1 = 0 (33 MHz clk) | PCB1 = 1 (66 MHz clk) | FAST_PCB1 = 1 (133 MHz clk) |
|---|---|---|
| 00 = CT 4 clocks, RP 6 clocks | 00 = **Reserved** | 00 = **Reserved** |
| 01 = CT 3 clocks, RP 5 clocks | 01 = CT 3 clocks, RP 8 clocks | 01 = CT 3 clocks, RP 16 clocks |
| 10 = CT 2 clocks, RP 4 clocks | 10 = CT 2 clocks, RP 8 clocks | 10 = **Reserved** |
| 11 = **Reserved** | 11 = **Reserved** | 11 = **Reserved** |

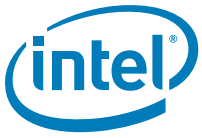
## 15.1.25 IDE_CONFIG—IDE I/O Configuration Register (SATA–D31:F2)

Address Offset: 54h–57h
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

***Note:*** This register is R/W to maintain software compatibility and enable parallel ATA functionality when the PCI functions are combined. These bits have no effect on SATA operation, unless otherwise noted.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** |
| 23:20 | Scratchpad (SP2). Chipset does not perform any actions on these bits. |
| 19:18 | **SEC_SIG_MODE** — R/W. These bits are used to control mode of the Secondary IDE signal pins for mobile swap bay support.<br>If the SRS bit (Chipset Config Registers:Offset 3414h:bit 1) is 1, the reset states of bits 19:18 will be 01 (tri-state) instead of 00 (normal).<br>00 = Normal (Enabled)<br>01 = Tri-state (Disabled)<br>10 = Drive low (Disabled)<br>11 = **Reserved** |
| 17:16 | **PRIM_SIG_MODE** — R/W. These bits are used to control mode of the Primary IDE signal pins for mobile swap bay support.<br>If the PRS bit (Chipset Config Registers:Offset 3414h:bit 1) is 1, the reset states of bits 17:16 will be 01 (tri-state) instead of 00 (normal).<br>00 = Normal (Enabled)<br>01 = Tri-state (Disabled)<br>10 = Drive low (Disabled)<br>11 = **Reserved** |

| Bit | Description |
|---|---|
| 15 | **Fast Secondary Drive 1 Base Clock (FAST_SCB1)** — R/W. This bit is used in conjunction with the SCT1 bits (D31:F2:4Ah, bits 13:12) to enable/disable Ultra ATA/100 timings for the Secondary Slave drive.<br>0 = Disable Ultra ATA/100 timing for the Secondary Slave drive.<br>1 = Enable Ultra ATA/100 timing for the Secondary Slave drive (overrides bit 3 in this register). |
| 14 | **Fast Secondary Drive 0 Base Clock (FAST_SCB0)** — R/W. This bit is used in conjunction with the SCT0 bits (D31:F2:4Ah, bits 9:8) to enable/disable Ultra ATA/100 timings for the Secondary Master drive.<br>0 = Disable Ultra ATA/100 timing for the Secondary Master drive.<br>1 = Enable Ultra ATA/100 timing for the Secondary Master drive (overrides bit 2 in this register). |
| 13 | **Fast Primary Drive 1 Base Clock (FAST_PCB1)** — R/W. This bit is used in conjunction with the PCT1 bits (D31:F2:4Ah, bits 5:4) to enable/disable Ultra ATA/100 timings for the Primary Slave drive.<br>0 = Disable Ultra ATA/100 timing for the Primary Slave drive.<br>1 = Enable Ultra ATA/100 timing for the Primary Slave drive (overrides bit 1 in this register). |
| 12 | **Fast Primary Drive 0 Base Clock (FAST_PCB0)** — R/W. This bit is used in conjunction with the PCT0 bits (D31:F2:4Ah, bits 1:0) to enable/disable Ultra ATA/100 timings for the Primary Master drive.<br>0 = Disable Ultra ATA/100 timing for the Primary Master drive.<br>1 = Enable Ultra ATA/100 timing for the Primary Master drive (overrides bit 0 in this register). |
| 11:8 | **Reserved** |
| 7:4 | Scratchpad (SP1). Chipset does not perform any action on these bits. |
| 3 | **Secondary Drive 1 Base Clock (SCB1)** — R/W.<br>0 = 33 MHz base clock for Ultra ATA timings.<br>1 = 66 MHz base clock for Ultra ATA timings |
| 2 | **Secondary Drive 0 Base Clock (SCBO)** — R/W.<br>0 = 33 MHz base clock for Ultra ATA timings.<br>1 = 66 MHz base clock for Ultra ATA timings |
| 1 | **Primary Drive 1 Base Clock (PCB1)** — R/W.<br>0 = 33 MHz base clock for Ultra ATA timings.<br>1 = 66 MHz base clock for Ultra ATA timings |
| 0 | **Primary Drive 0 Base Clock (PCB0)** — R/W.<br>0 = 33 MHz base clock for Ultra ATA timings.<br>1 = 66 MHz base clock for Ultra ATA timings |



## 15.1.26 PID—PCI Power Management Capability Identification Register (SATA–D31:F2)

Address Offset: 70h–71h Attribute: RO
Default Value: XX01h Size: 16 bits

| Bits | Description |
|---|---|
| 15:8 | Next Capability (NEXT) — RO.<br>00h — if SCC = 01h (IDE mode).<br>A8h — for all other values of SCC to point to the next capability structure.<br>This field is changed to 00h if the SCRD bit (D31:F2;Offset 94h bit-30) is set. |
| 7:0 | Capability ID (CID) — RO. Indicates that this pointer is a PCI power management. |

## 15.1.27 PC—PCI Power Management Capabilities Register (SATA–D31:F2)

Address Offset: 72h–73h Attribute: RO
Default Value: 4002h Size: 16 bits

| Bits | Description |
|---|---|
| 15:11 | PME Support (PME_SUP) — RO. Indicates PME# can be generated from the $D3_{HOT}$ state in the SATA host controller. |
| 10 | D2 Support (D2_SUP) — RO. Hardwired to 0. The D2 state is not supported |
| 9 | D1 Support (D1_SUP) — RO. Hardwired to 0. The D1 state is not supported |
| 8:6 | Auxiliary Current (AUX_CUR) — RO. PME# from D3COLD state is not supported, therefore this field is 000b. |
| 5 | Device Specific Initialization (DSI) — RO. Hardwired to 0 to indicate that no device-specific initialization is required. |
| 4 | **Reserved** |
| 3 | PME Clock (PME_CLK) — RO. Hardwired to 0 to indicate that PCI clock is not required to generate PME#. |
| 2:0 | Version (VER) — RO. Hardwired to 010 to indicates support for Revision 1.1 of the PCI Power Management Specification. |

## 15.1.28 PMCS—PCI Power Management Control and Status Register (SATA–D31:F2)

Address Offset: 74h–75h Attribute: RO, R/W, R/WC
Default Value: 0000h Size: 16 bits

| Bits | Description |
|---|---|
| 15 | **PME Status (PMES)** — R/WC. Bit is set when a PME event is to be requested, and if this bit and PMEE is set, a PME# will be generated from the SATA controller. |
| 14:9 | **Reserved** |

| Bits | Description |
|---|---|
| 8 | **PME Enable (PMEE)** — R/W.<br>0 = Disable.<br>1 = Enable. SATA controller generates PME# form $D3_{HOT}$ on a wake event. |
| 7:2 | **Reserved** |
| 1:0 | **Power State (PS)** — R/W. These bits are used both to determine the current power state of the SATA controller and to set a new power state.<br>00 = D0 state<br>11 = $D3_{HOT}$ state<br>When in the $D3_{HOT}$ state, the controller's configuration space is available, but the I/O and memory spaces are not. Additionally, interrupts are blocked. |

## 15.1.29 MSICI—Message Signaled Interrupt Capability Identification (SATA–D31:F2)

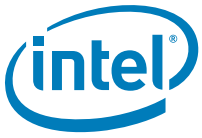
Address Offset: 80h–81h
Default Value: 7005h
Attribute: RO
Size: 16 bits

| Bits | Description |
|---|---|
| 15:8 | **Next Pointer (NEXT):** Indicates the next item in the list is the PCI power management pointer. |
| 7:0 | **Capability ID (CID):** Capabilities ID indicates MSI. |

## 15.1.30 MSIMC—Message Signaled Interrupt Message Control (SATA–D31:F2)

Address Offset: 82h–83h
Default Value: 0000h
Attribute: R/W, RO
Size: 16 bits

| Bits | Description |
|---|---|
| 15:8 | **Reserved** |
| 7 | **64 Bit Address Capable (C64):** Capable of generating a 32-bit message only. |



<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>6:4</td><td>Multiple Message Enable (MME): When this field is cleared to ‘000’ (and MSIE is set), only a single MSI message will be generated for all SATA ports, and bits [15:0] of the message vector will be driven from MD[15:0].<br>When this field is set to ‘001’ (and MSIE is set), two MSI messages will be generated. Bit [15:1] of the message vectors will be driven from MD[15:1] and bit [0] of the message vector will be driven dependent on which SATA port is the source of the interrupt: ‘0’ for port 0, and ‘1’ for ports 1, 2 and 3.<br>When this field is set to ‘010’ (and MSIE is set), four message will be generated, one for each SATA port. Bits[15:2] of the message vectors will be driven from MD[15:2], while bits[1:0] will be driven dependent on which SATA port is the source of the interrupt: ‘00’ for port 0, ‘01’ for port 1, ‘10’ for port 2, and ‘11’ for port 3.
<table>
<tr><th rowspan="2">MME</th><th colspan="3">Value Driven on MSI Memory Write</th></tr>
<tr><th>Bits[15:2]</th><th>Bit[1]</th><th>Bit[0]</th></tr>
<tr><td>000</td><td>MD[15:2]</td><td>MD[1]</td><td>MD[0]</td></tr>
<tr><td>001</td><td>MD[15:2]</td><td>MD[1]</td><td>Ports 0: 0<br>Ports 1,2,3: 1</td></tr>
<tr><td>010</td><td>MD[15:2]</td><td>Port 0: 0<br>Port 1: 0<br>Port 2: 1<br>Port 3: 1</td><td>Port 0: 0<br>Port 1: 1<br>Port 2: 0<br>Port 3: 1</td></tr>
<tr><td>011–111</td><td colspan="3">Reserved</td></tr>
</table>
Values ‘011b’ to ‘111b’ are reserved. If this field is set to one of these reserved values, the results are undefined.</td></tr>
<tr><td>3:1</td><td>Multiple Message Capable (MMC): Indicates that the Chipset SATA controller supports four interrupt messages.</td></tr>
<tr><td>0</td><td>MSI Enable (MSIE): If set, MSI is enabled and traditional interrupt pins are not used to generate interrupts.</td></tr>
</table>

## 15.1.31 MSIMA— Message Signaled Interrupt Message Address (SATA–D31:F2)

Address Offset: 84h–87h
Default Value: 00000000h
Attribute: RO, R/W
Size: 32 bits

| Bits | Description |
|---|---|
| 31:2 | **Address (ADDR):** Lower 32 bits of the system specified message address, always DWord aligned. |
| 1:0 | **Reserved** |

### 15.1.32 MSIMD—Message Signaled Interrupt Message Data (SATA–D31:F2)

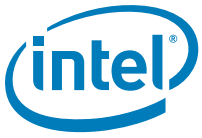
Address Offset: 88h-89h
Default Value: 0000h
Attribute: R/W
Size: 16 bits

| Bits | Description |
|---|---|
| 15:0 | **Data (DATA)**— R/W**:** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the lower word of the data bus of the MSI memory write transaction. Note that when the MME field is set to '001' or '010', bit [0] and bits [1:0] respectively of the MSI memory write transaction will be driven based on the source of the interrupt rather than from MD[1:0]. See the description of the MME field. |

### 15.1.33 MAP—Address Map Register (SATA–D31:F2)

Address Offset: 90h
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bits | Description |
|---|---|
| 7:6 | **SATA Mode Select (SMS)** — R/W: Software programs these bits to control the mode in which the SATA HBA should operate:<br>00b = IDE mode<br>01b = AHCI mode<br>10b = **Reserved**<br>11b = **Reserved**<br><br>**NOTES:**<br>1. The SATA Function Device ID will change based on the value of this register.<br>2. When combined mode is used (MV Not equal to '0'), only IDE mode is allowed. IDE mode can be selected when AHCI are enabled<br>3. AHCI mode may only be selected when MV = 0<br>4. Programming these bits with values that are invalid (e.g, selecting RAID when in combined mode) will result in indeterministic behavior by the hardware. |
| 5:2 | **Reserved**. |
| 1:0 | **Map Value (MV)**: The value in the bits below indicate the address range the SATA ports. BIOS must programs it to value 00, in order to set the Port 0 as primary and Port 1 as secondary. |



## 15.1.34 PCS—Port Control and Status Register (SATA–D31:F2)

| | | | |
|---|---|---|---|
| Address Offset: | 92h–93h | Attribute: | R/W, R/WC, RO |
| Default Value: | 0000h | Size: | 16 bits |

This register is only used in systems that do not support AHCI. In AHCI enabled systems, bits[3:0] must always be set bits[2,0] and the status of the port is controlled through AHCI memory space.

| Bits | Description |
|---|---|
| 15:6 | **Reserved**. |
| 5 | **Port 1 Present (P1P)** — RO. The status of this bit may change at any time. This bit is cleared when the port is disabled via P1E. This bit is not cleared upon surprise removal of a device.<br>0 = No device detected.<br>1 = The presence of a device on Port 1 has been detected. |
| 4 | **Port 0 Present (P0P)** — RO. The status of this bit may change at any time. This bit is cleared when the port is disabled via P0E. This bit is not cleared upon surprise removal of a device.<br>0 = No device detected.<br>1 = The presence of a device on Port 0 has been detected. |
| 3:2 | **Reserved**. |
| 1 | **Port 1 Enabled (P1E)** — R/W.<br>0 = Disabled. The port is in the ‘off’ state and cannot detect any devices.<br>1 = Enabled. The port can transition between the on, partial, and slumber states and can detect devices.<br>**NOTE:** This bit takes precedence over P1CMD.SUD (offset ABAR+198h:bit 1) |
| 0 | **Port 0 Enabled (P0E)** — R/W.<br>0 = Disabled. The port is in the ‘off’ state and cannot detect any devices.<br>1 = Enabled. The port can transition between the on, partial, and slumber states and can detect devices.<br>**NOTE:** This bit takes precedence over P0CMD.SUD (offset ABAR+118h:bit 1) |

## 15.1.35 SIR—SATA Initialization Register

Address Offset: 94h–97h Attribute: R/W
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:29 | **Reserved** |
| 30 | **SATA Capability Registers Disable (SCRD)**.<br>When this bit is set, the SATA Capability Registers are disabled. That is, SATA Capability Registers 0 and 1 are both changed to Read Only with the value of 00000000h. Also, the Next Capability bits in the PCI Power Management Capability Information Register (D31:F2;Offset 70h bits 15:8) are changed to 00h, to indicate that the PCI Power Management Capability structure is the last PCI capability structure in the SATA controller. When this bit is cleared, the SATA Capability Registers are enabled. |
| 29:28 | **Reserved** |
| 27:24 | SATA Initialization Field 3 (SIF3) — R/W. BIOS shall always program this register to the value 0Ch. All other values are reserved. |
| 23 | SATA Initialization Field 2 (SIF2) — R/W. BIOS shall always program this register to the value 1b. All other values are reserved. |
| 22:10 | **Reserved** |
| 9 | SCR Access Enable (SCRAE) — R/W. In non-AHCI mode, this bit allows access to the SATA SCR registers (SStatus, SControl, and SError registers).<br>0 = The ABAR (Dev31:F2:Offset 24h) register and MSE bit field (Dev31:F2:Offset 04h:bit 1) remain as defined.<br>1 = The ABAR (Dev31:F2:Offset 24h) register and MSE bit field (Dev31:F2:Offset 04h:bit 1) are forced to be read/write.<br><br>**NOTES:**<br>1. Using this mode only allows access to AHCI registers PxSSTS, PxSCTL, PxSERR. All other AHCI space is reserved when this bit is set.<br>2. Proper use of this bit requires:<br>• ABAR must be programmed to a valid BAR; MSE must be set before software can access AHCI space.<br>• The Port Implemented bit (D31:F2, Offset ABAR+0Ch) for the corresponding port has to be set to allow access to the AHCI port specific PxSSTS, PxSCTL, and PxSERR registers. |
| 8:0 | SATA Initialization Field 1 (SIF1) — R/W. BIOS shall always program this register to the value 180h. All other values are reserved. |

## 15.1.36 SIRI—SATA Indexed Registers Index

Address Offset: A0h Attribute: R/W
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:2 | Index (IDX) — R/W. This field contains a 6-bit index pointer into the SATA Indexed Register space. Data is written into and read from the SIRD register (D31:F2:A4h). |
| 1:0 | **Reserved** |

SATA Indexed Registers

| Index | Name |
|---|---|
| 00h–03h | SATA TX Termination Test Register 1 (STTT1) |
| 04h–1Bh | **Reserved** |
| 1Ch–1Fh | SATA Test Mode Enable Register (STME) |
| 20h–73h | **Reserved** |
| 74h–77h | SATA TX Termination Test Register 2 (STTT2) |
| 78h–FFh | **Reserved** |

## 15.1.37 STRD—SATA Indexed Register Data

Address Offset: A4h Attribute: R/W
Default Value: XXXXXXXXh Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | Data (DTA) — R/W. This field contains a 32-bit data value that is written to the register pointed to by SIRI (D31:F2;A0h) or read from the register pointed to by SIRI. |

### 15.1.37.1 STTT1—SATA Indexed Registers Index 00h (SATA TX Termination Test Register 1)

Address Offset: Index 00h–03h Attribute: R/W
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:2 | **Reserved**. |
| 1 | **Port 1 TX Termination Test Enable** — R/W:<br>0 = Port 1 TX termination port testing is disabled.<br>1 = Enables testing of Port 1 TX termination.<br><br>**NOTE:** This bit only to be used for system board testing. |
| 0 | **Port 0 TX Termination Test Enable** — R/W:<br>0 = Port 0 TX termination port testing is disabled.<br>1 = Enables testing of Port 0 TX termination.<br><br>**NOTE:** This bit only to be used for system board testing. |

### 15.1.37.2 STME—SATA Indexed Registers Index C1h (SATA Test Mode Enable Register)

Address Offset: Index 1Ch–1Fh Attribute: R/W
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:19 | **Reserved**. |
| 18 | **SATA Test Mode Enable Bit** — R/W:<br>0 = Entrance to Chipset SATA test modes are disabled.<br>1 = This bit allows entrance to Chipset SATA test modes when set.<br><br>**NOTE:** This bit only to be used for system board testing. |
| 17:0 | **Reserved**. |

### 15.1.37.3 STTT2 — SATA Indexed Registers Index 74h (SATA TX Termination Test Register 2)

Address Offset: Index 74h – 77h Attribute: R/W
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:18 | **Reserved**. |
| 17 | **Port 3 TX Termination Test Enable** — R/W:<br>0 = Port 3 TX termination port testing is disabled.<br>1 = Enables testing of Port 3 TX termination.<br><br>**NOTE:** This bit only to be used for system board testing. |
| 16 | **Port 2 TX Termination Test Enable** — R/W:<br>0 = Port 2 TX termination port testing is disabled.<br>1 = Enables testing of Port 2 TX termination.<br><br>**NOTE:** This bit only to be used for system board testing. |
| 15:0 | **Reserved**. |

## 15.1.38 SCAP0—SATA Capability Register 0 (SATA–D31:F2)

Address Offset: A8h–ABh Attribute: RO
Default Value: 00100012h Size: 32 bits

This register is set to 00000000h if the SCRD bit (D31:F2;Offset 94h bit-30) is set.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** |
| 23:20 | Major Revision (MAJREV) — RO: Major revision number of the SATA Capability Pointer implemented. |
| 19:16 | Minor Revision (MINREV) — RO: Minor revision number of the SATA Capability Pointer implemented. |

| Bit | Description |
|---|---|
| 15:8 | Next Capability Pointer (NEXT) — RO: Points to the next capability structure. 00h indicates this is the last capability pointer. |
| 7:0 | Capability ID (CAP)— RO: This value of 12h has been assigned by the PCI SIG to designate the SATA Capability Structure. |

## 15.1.39 SCAP1—SATA Capability Register 1 (SATA–D31:F2)

Address Offset: ACh–AFh Attribute: RO
Default Value: 00000048h Size: 32 bits

This register is set to 00000000h if the SCRD bit (D31:F2;Offset 94h bit-30) is set.

| Bit | Description |
|---|---|
| 31:16 | **Reserved** |
| 15:4 | BAR Offset (BAROFST) — RO: Indicates the offset into the BAR where the Index/Data pair are located (in DWord granularity). The Index and Data I/O registers are located at offset 10h within the I/O space defined by LBAR. A value of 004h indicates offset 10h.<br>000h = 0h offset<br>001h = 4h offset<br>002h = 8h offset<br>003h = Bh offset<br>004h = 10h offset<br>...<br>FFFh = 3FFFh offset (max 16KB) |
| 3:0 | BAR Location (BARLOC) — RO: Indicates the absolute PCI Configuration Register address of the BAR containing the Index/Data pair (in DWord granularity). The Index and Data I/O registers reside within the space defined by LBAR in the SATA controller. A value of 8h indicates offset 20h, which is LBAR.<br>0000 – 0011b = reserved<br>0100b = 10h => BAR0<br>0101b = 14h => BAR1<br>0110b = 18h => BAR2<br>0111b = 1Ch => BAR3<br>1000b = 20h => LBAR<br>1001b = 24h => BAR5<br>1010 – 1110b = reserved<br>1111b = Index/Data pair in PCI Configuration space. This is not supported in Chipset. |

### 15.1.40 ATC—APM Trapping Control Register (SATA–D31:F2)

Address Offset: C0h
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bit | Description |
|---|---|
| 7:4 | **Reserved** |
| 3 | **Secondary Slave Trap (SST)** — R/W. Enables trapping and SMI# assertion on legacy I/O accesses to 170h–177h and 376h. The active device on the secondary interface must be device 1 for the trap and/or SMI# to occur. |
| 2 | **Secondary Master Trap (SMT)** — R/W. Enables trapping and SMI# assertion on legacy I/O accesses to 170h–177h and 376h. The active device on the secondary interface must be device 0 for the trap and/or SMI# to occur. |
| 1 | **Primary Slave Trap (PST)** — R/W. Enables trapping and SMI# assertion on legacy I/O accesses to 1F0h–1F7h and 3F6h. The active device on the primary interface must be device 1 for the trap and/or SMI# to occur. |
| 0 | **Primary Master Trap (PMT)** — R/W. Enables trapping and SMI# assertion on legacy I/O accesses to 1F0h–1F7h and 3F6h. The active device on the primary interface must be device 0 for the trap and/or SMI# to occur. |

### 15.1.41 ATS—APM Trapping Status Register (SATA–D31:F2)

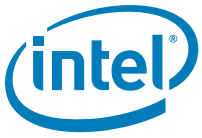
Address Offset: C4h
Default Value: 00h
Attribute: R/WC
Size: 8 bits

| Bit | Description |
|---|---|
| 7:4 | **Reserved** |
| 3 | **Secondary Slave Trap (SST)** — R/WC. Indicates that a trap occurred to the secondary slave device. |
| 2 | **Secondary Master Trap (SPT)** — R/WC. Indicates that a trap occurred to the secondary master device. |
| 1 | **Primary Slave Trap (PST)** — R/WC. Indicates that a trap occurred to the primary slave device. |
| 0 | **Primary Master Trap (PMT)** — R/WC. Indicates that a trap occurred to the primary master device. |

### 15.1.42 SP — Scratch Pad Register (SATA–D31:F2)

Address Offset: D0h
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Data (DT)** — R/W. This is a read/write register that is available for software to use. No hardware action is taken on this register. |



## 15.1.43 BFCS—BIST FIS Control/Status Register (SATA–D31:F2)

Address Offset: E0h–E3h
Default Value: 00000000h
Attribute: R/W, R/WC
Size: 32 bits

| Bits | Description |
|---|---|
| 31:12 | **Reserved** |
| 11 | **BIST FIS Successful (BFS)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = BIST FIS transmitted by Chipset received an R_OK completion status from the device.<br><br>**NOTE:** This bit must be cleared by software prior to initiating a BIST FIS. |
| 10 | **BIST FIS Failed (BFF)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = BIST FIS transmitted by Chipset received an R_ERR completion status from the device.<br><br>**NOTE:** This bit must be cleared by software prior to initiating a BIST FIS. |
| 9 | **Port 1 BIST FIS Initiate (P1BFI)** — R/W. When a rising edge is detected on this bit field, the Chipset initiates a BIST FIS to the device on Port 1, using the parameters specified in this register and the data specified in BFTD1 and BFTD2. The BIST FIS will only be initiated if a device on Port 1 is present and ready (not partial/slumber state). After a BIST FIS is successfully completed, software must disable and re-enable the port using the PxE bits at offset 92h prior to attempting additional BIST FISes or to return the Chipset to a normal operational mode. If the BIST FIS fails to complete, as indicated by the BFF bit in the register, then software can clear then set the P1BFI bit to initiate another BIST FIS. This can be retried until the BIST FIS eventually completes successfully |
| 8 | **Port 0 BIST FIS Initiate (P0BFI)** — R/W. When a rising edge is detected on this bit field, the Chipset initiates a BIST FIS to the device on Port 0, using the parameters specified in this register and the data specified in BFTD1 and BFTD2. The BIST FIS will only be initiated if a device on Port 0 is present and ready (not partial/slumber state). After a BIST FIS is successfully completed, software must disable and re-enable the port using the PxE bits at offset 92h prior to attempting additional BIST FISes or to return the Chipset to a normal operational mode. If the BIST FIS fails to complete, as indicated by the BFF bit in the register, then software can clear then set the P0BFI bit to initiate another BIST FIS. This can be retried until the BIST FIS eventually completes successfully |
| 7:2 | **BIST FIS Parameters**. These 6 bits form the contents of the upper 6 bits of the BIST FIS Pattern Definition in any BIST FIS transmitted by the Chipset. This field is not port specific; its contents will be used for any BIST FIS initiated on port 0, port 1, port 2 or port 3. The specific bit definitions are:<br>Bit 7: T – Far End Transmit mode<br>Bit 6: A – Align Bypass mode<br>Bit 5: S – Bypass Scrambling<br>Bit 4: L – Far End Retimed Loopback<br>Bit 3: F – Far End Analog Loopback<br>Bit 2: P – Primitive bit for use with Transmit mode |
| 1:0 | **Reserved** |

### 15.1.44 BFTD1—BIST FIS Transmit Data1 Register (SATA–D31:F2)

Address Offset: E4h–E7h
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

| Bits | Description |
|---|---|
| 31:0 | **BIST FIS Transmit Data 1** — R/W. The data programmed into this register will form the contents of the second DWord of any BIST FIS initiated by the Chipset. This register is not port specific; its contents will be used for BIST FIS initiated on any port. Although the 2nd and 3rd DWs of the BIST FIS are only meaningful when the “T” bit of the BIST FIS is set to indicate “Far-End Transmit mode”, this register’s contents will be transmitted as the BIST FIS 2nd DW regardless of whether or not the “T” bit is indicated in the BFCS register (D31:F2:E0h). |

### 15.1.45 BFTD2—BIST FIS Transmit Data2 Register (SATA–D31:F2)

Address Offset: E8h–EBh
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

| Bits | Description |
|---|---|
| 31:0 | **BIST FIS Transmit Data 2** — R/W. The data programmed into this register will form the contents of the third DWord of any BIST FIS initiated by the Chipset. This register is not port specific; its contents will be used for BIST FIS initiated on any port. Although the 2nd and 3rd DWs of the BIST FIS are only meaningful when the “T” bit of the BIST FIS is set to indicate “Far-End Transmit mode”, this register’s contents will be transmitted as the BIST FIS 3rd DW regardless of whether or not the “T” bit is indicated in the BFCS register (D31:F2:E0h). |

## 15.2 Bus Master IDE I/O Registers (D31:F2)

The bus master IDE function uses 16 bytes of I/O space, allocated via the BAR register, located in Device 31:Function 2 Configuration space, offset 20h. All bus master IDE I/O space registers can be accessed as byte, word, or DWord quantities. Reading reserved bits returns an indeterminate, inconsistent value, and writes to reserved bits have no effect (but should not be attempted). These registers are only used for legacy operation. Software must not use these registers when running AHCI. The description of the I/O registers is shown in Table 15-136.

**Table 15-136.Bus Master IDE I/O Register Address Map**

| BAR+ Offset | Mnemonic | Register | Default | Type |
|---|---|---|---|---|
| 00 | BMICP | Command Register Primary | 00h | R/W |
| 01 | — | **Reserved** | — | RO |
| 02 | BMISP | Bus Master IDE Status Register Primary | 00h | R/W, R/WC, RO |
| 03 | — | **Reserved** | — | RO |

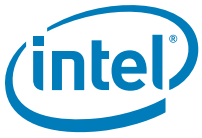


**Table 15-136.Bus Master IDE I/O Register Address Map**

| BAR+ Offset | Mnemonic | Register | Default | Type |
|---|---|---|---|---|
| 04–07 | BMIDP | Bus Master IDE Descriptor Table Pointer Primary | xxxxxxxxh | R/W |
| 08 | BMICS | Command Register Secondary | 00h | R/W |
| 09 | — | **Reserved** | — | RO |
| 0Ah | BMISS | Bus Master IDE Status Register Secondary | 00h | R/W, R/WC, RO |
| 0Bh | — | **Reserved** | — | RO |
| 0Ch–0Fh | BMIDS | Bus Master IDE Descriptor Table Pointer Secondary | xxxxxxxxh | R/W |

## 15.2.1 BMIC[P,S]—Bus Master IDE Command Register (D31:F2)

Address Offset: Primary: BAR + 00h
Secondary: BAR + 08h
Attribute: R/W
Default Value: 00h
Size: 8 bits

| Bit | Description |
|---|---|
| 7:4 | **Reserved**. Returns 0. |
| 3 | **Read / Write Control (R/WC)** — R/W. This bit sets the direction of the bus master transfer: This bit must NOT be changed when the bus master function is active.<br>0 = Memory reads<br>1 = Memory writes |
| 2:1 | **Reserved**. Returns 0. |
| 0 | **Start/Stop Bus Master (START)** — R/W.<br>0 = All state information is lost when this bit is cleared. Master mode operation cannot be stopped and then resumed. If this bit is reset while bus master operation is still active (i.e., the Bus Master IDE Active bit (D31:F2:BAR + 02h, bit 0) of the Bus Master IDE Status register for that IDE channel is set) and the drive has not yet finished its data transfer (the Interrupt bit in the Bus Master IDE Status register for that IDE channel is not set), the bus master command is said to be aborted and data transferred from the drive may be discarded instead of being written to system memory.<br>1 = Enables bus master operation of the controller. Bus master operation does not actually start unless the Bus Master Enable bit (D31:F1:04h, bit 2) in PCI configuration space is also set. Bus master operation begins when this bit is detected changing from 0 to 1. The controller will transfer data between the IDE device and memory only when this bit is set. Master operation can be halted by writing a 0 to this bit.<br><br>**NOTE:** This bit is intended to be cleared by software after the data transfer is completed, as indicated by either the Bus Master IDE Active bit being cleared or the Interrupt bit of the Bus Master IDE Status register for that IDE channel being set, or both. Hardware does not clear this bit automatically. If this bit is cleared to 0 prior to the DMA data transfer being initiated by the drive in a |

### 15.2.2 BMIS[P,S]—Bus Master IDE Status Register (D31:F2)

Address Offset: Primary: BAR + 02h
Secondary: BAR + 0Ah
Attribute: R/W, R/WC, RO
Default Value: 00h
Size: 8 bits

| Bit | Description |
|---|---|
| 7 | **PRD Interrupt Status (PRDIS)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit is set when the host controller execution of a PRD that has its PRD_INT bit set. |
| 6 | **Drive 1 DMA Capable** — R/W.<br>0 = Not Capable.<br>1 = Capable. Set by device dependent code (BIOS or device driver) to indicate that drive 1 for this channel is capable of DMA transfers, and that the controller has been initialized for optimum performance. The Chipset does not use this bit. It is intended for systems that do not attach BMIDE to the PCI bus. |
| 5 | **Drive 0 DMA Capable** — R/W.<br>0 = Not Capable<br>1 = Capable. Set by device dependent code (BIOS or device driver) to indicate that drive 0 for this channel is capable of DMA transfers, and that the controller has been initialized for optimum performance. The Chipset does not use this bit. It is intended for systems that do not attach BMIDE to the PCI bus. |
| 4:3 | **Reserved**. Returns 0. |
| 2 | **Interrupt** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Set when a device FIS is received with the 'I' bit set, provided that software has not disabled interrupts via the IEN bit of the Device Control Register (see chapter 5 of the *Serial ATA Specification*, Revision 1.0a). |
| 1 | **Error** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit is set when the controller encounters a target abort or master abort when transferring data on PCI. |
| 0 | **Bus Master IDE Active (ACT)** — RO.<br>0 = This bit is cleared by the Chipset when the last transfer for a region is performed, where EOT for that region is set in the region descriptor. It is also cleared by the Chipset when the Start Bus Master bit (D31:F2:BAR+ 00h, bit 0) is cleared in the Command register. When this bit is read as a 0, all data transferred from the drive during the previous bus master command is visible in system memory, unless the bus master command was aborted.<br>1 = Set by the Chipset when the Start bit is written to the Command register. |

### 15.2.3 BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register (D31:F2)

Address Offset: Primary: BAR + 04h–07h Attribute: R/W
Secondary: BAR + 0Ch–0Fh
Default Value: All bits undefined Size: 32 bits

| Bit | Description |
|---|---|
| 31:2 | **Address of Descriptor Table (ADDR)** — R/W. The bits in this field correspond to A[31:2]. The Descriptor Table must be DWord-aligned. The Descriptor Table must not cross a 64-K boundary in memory. |
| 1:0 | **Reserved** |

## 15.3 AHCI Registers (D31:F2)

***Note:*** These registers are AHCI-specific and available only on Chipset components that support AHCI (not on the 82801GB Chipset) and when the Chipset is properly configured. The Serial ATA Status, Control, and Error registers are special exceptions and may be accessed on all Chipset components if properly configured (see section Section 15.1.35 for details).

The memory mapped registers within the SATA controller exist in non-cacheable memory space. Additionally, locked accesses are not supported. If software attempts to perform locked transactions to the registers, indeterminate results may occur. Register accesses shall have a maximum size of 64-bits; 64-bit access must not cross an 8-byte alignment boundary.

The registers are divided into two sections – generic host control and port control. The port control registers are the same for all ports, and there are as many registers banks as there are ports.

**Table 15-137.AHCI Register Address Map**

| ABAR + Offset | Mnemonic | Register |
|---|---|---|
| 00–1Fh | GHC | Generic Host Control |
| 20h–FFh | — | **Reserved** |
| 100h–17Fh | P0PCR | Port 0 port control registers |
| 180h–1FFh | P1PCR | Port 1 port control registers |
| 200h–3FFh | — | **Reserved** |

## 15.3.1 AHCI Generic Host Control Registers (D31:F2)

**Table 15-138.Generic Host Controller Register Address Map**

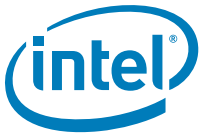
| ABAR + Offset | Mnemonic | Register | Default | Type |
|---|---|---|---|---|
| 00–03 | CAP | Host Capabilities | DE22FF03h | R/WO, RO |
| 04–07 | GHC | Global Chipset Control | 00000000h | R/W |
| 08–0Bh | IS | Interrupt Status | 00000000h | R/WC, RO |
| 0Ch–0Fh | PI | Ports Implemented | 00000000h | R/WO, RO |
| 10h–13h | VS | AHCI Version | 00010100h | RO |

### 15.3.1.1 CAP—Host Capabilities Register (D31:F2)

Address Offset: ABAR + 00h–03h
Attribute: R/WO, RO
Default Value: DE22FF03h
Size: 32 bits

All bits in this register that are R/WO are reset only by PLTRST#.

| Bit | Description |
|---|---|
| 31 | **Supports 64-bit Addressing (S64A)** — RO. Indicates that the SATA controller can access<br>64-bit data structures. The 32-bit upper bits of the port DMA Descriptor, the PRD Base, and each PRD entry are read/write. |
| 30 | **Supports Command Queue Acceleration (SCQA)** — RO. Hardwired to 1 to indicate that the SATA controller supports SATA command queuing via the DMA Setup FIS. The Chipset handles DMA Setup FISes natively, and can handle auto-activate optimization through that FIS. |
| 29 | Supports SNotification Register (SSNTF): — RO. The Chipset SATA Controller does not support the SNotification register. |
| 28 | **Supports Interlock Switch (SIS)** — R/WO. Indicates whether the SATA controller supports interlock switches on its ports for use in Hot-Plug operations. This value is loaded by platform BIOS prior to OS initialization.<br>If this bit is set, BIOS must also map the SATAGP pins to the SATA controller through GPIO space. |
| 27 | **Supports Staggered Spin-up (SSS)** — R/WO. Indicates whether the SATA controller supports staggered spin-up on its ports, for use in balancing power spikes. This value is loaded by platform BIOS prior to OS initialization.<br>0 = Staggered spin-up not supported.<br>1 = Staggered spin-up supported. |
| 26 | **Supports Aggressive Link Power Management (SALP)** — R/WO. Indicates the SATA controller supports auto-generating link requests to the partial or slumber states when there are no commands to process.<br>0 = Aggressive link power management not supported.<br>1 = Aggressive link power management supported. |
| 25 | **Supports Activity LED (SAL)** — RO. Indicates that the SATA controller supports a single output pin (SATALED#) which indicates activity. |



| Bit | Description |
|---|---|
| 24 | Supports Command List Override (SCLO) — R/WO. When set to 1, indicates that the HBA supports the PxCMD.CLO bit and it's associated function. When cleared to '0', The HBA is not capable of clearing the BSY and DRQ bits in the Status register in order to issue a software reset if these bits are still set from a previous operation. |
| 23:20 | Interface Speed Support (ISS) — R/WO. Indicates the maximum speed the SATA controller can support on its ports.<br>2h =3.0 Gb/s. |
| 19 | Supports Non-Zero DMA Offsets (SNZO) — RO. **Reserved**, as per the AHCI Revision 1.0 specification |
| 18 | Supports Port Selector Acceleration — RO. Port Selectors not supported. |
| 17 | **Supports Port Multiplier (PMS)** — R/WO. Chipset does not support port multiplier. BIOS/SW shall write this bit to '0' during AHCI initialization. |
| 16 | Supports Port Multiplier FIS Based Switching (PMFS) — RO. **Reserved**, as per the AHCI Revision 1.0 specification.<br><br>**NOTE:** Port Multiplier not supported by Chipset. |
| 15 | PIO Multiple DRQ Block (PMD) — R/WO. The SATA controller supports PIO Multiple DRQ Command Block |
| 14 | Slumber State Capable (SSC) — RO. The SATA controller supports the slumber state. |
| 13 | Partial State Capable (PSC) — RO. The SATA controller supports the partial state. |
| 12:8 | Number of Command Slots (NCS) — RO. Hardwired to 1Fh to indicate support for 32 slots. |
| 7:5 | **Reserved**. Returns 0. |
| 4:0 | Number of Ports (NPS) — RO. Hardwired to 3h to indicate support for 4 ports. Note that the number of ports indicated in this field may be more than the number of ports indicated in the PI (ABAR + 0Ch) register. |

### 15.3.1.2 GHC—Global Chipset Control Register (D31:F2)

Address Offset: ABAR + 04h–07h Attribute: R/W
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31 | **AHCI Enable (AE)** — R/W. When set, indicates that an AHCI driver is loaded and the controller will be talked to via AHCI mechanisms. This can be used by an Chipset that supports both legacy mechanisms (such as SFF-8038i) and AHCI to know when the controller will not be talked to as legacy.<br>When set, software will only talk to the Chipset using AHCI. The Chipset will not have to allow command processing via both AHCI and legacy mechanisms. When cleared, software will only talk to the Chipset using legacy mechanisms.<br>Software shall set this bit to 1 before accessing other AHCI registers. |

| Bit | Description |
|---|---|
| 30:2 | **Reserved**. Returns 0. |
| 1 | **Interrupt Enable (IE)** — R/W. This global bit enables interrupts from the Chipset.<br>0 = All interrupt sources from all ports are disabled.<br>1 = Interrupts are allowed from the AHCI controller. |
| 0 | **HBA Reset (HR)** — R/W. Resets Chipset AHCI controller.<br>0 = No effect<br>1 = Causes an internal reset of the Chipset AHCI controller. All state machines that relate to data transfers and queuing return to an idle condition, and all ports are re-initialized via COMRESET.<br><br>**NOTE:** For further details, consult section 12.3.3 of the *Serial ATA Advanced Host Controller Interface* specification. |

### 15.3.1.3 IS—Interrupt Status Register (D31:F2)

Address Offset: ABAR + 08h–0Bh Attribute: R/WC, RO
Default Value: 00000000h Size: 32 bits

This register indicates which of the ports within the controller have an interrupt pending and require service.

| Bit | Description |
|---|---|
| 31:2 | **Reserved**. Returns 0. |
| 1 | **Interrupt Pending Status Port[1] (IPS[1])** — R/WC.<br>0 = No interrupt pending.<br>1 = Port 1has an interrupt pending. Software can use this information to determine which ports require service after an interrupt. |
| 0 | **Interrupt Pending Status Port[0] (IPS[0])** — R/WC.<br>0 = No interrupt pending.<br>1 = Port 0 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt. |

### 15.3.1.4 PI—Ports Implemented Register (D31:F2)

Address Offset: ABAR + 0Ch–0Fh Attribute: R/WO, RO
Default Value: 00000000h Size: 32 bits

This register indicates which ports are exposed to the Chipset. It is loaded by platform BIOS. It indicates which ports that the device supports are available for software to use. For ports that are not available, software must not read or write to registers within that port.

| Bit | Description |
|---|---|
| 31:2 | **Reserved**. Returns 0. |
| 1 | **Ports Implemented Port 1 (PI1)** — R/WO.<br>0 = The port is not implemented.<br>1 = The port is implemented. |

| Bit | Description |
|---|---|
| 0 | **Ports Implemented Port 0 (PI0)** — R/WO.<br>0 = The port is not implemented.<br>1 = The port is implemented. |

#### 15.3.1.5 VS—AHCI Version (D31:F2)

Address Offset: ABAR + 10h–13h Attribute: RO
Default Value: 00010100h Size: 32 bits

This register indicates the major and minor version of the AHCI specification. It is BCD encoded. The upper two bytes represent the major version number, and the lower two bytes represent the minor version number. Example: Version 3.12 would be represented as 00030102h. The current version of the specification is 1.10 (00010100h).

| Bit | Description |
|---|---|
| 31:16 | Major Version Number (MJR) — RO. Indicates the major version is 1 |
| 15:0 | Minor Version Number (MNR) — RO. Indicates the minor version is 10. |

### 15.3.2 Port Registers (D31:F2)

**Table 15-139.Port [1:0] DMA Register Address Map (Sheet 1 of 2)**

| ABAR + Offset | Mnemonic | Register |
|---|---|---|
| 100h–103h | P0CLB | Port 0 Command List Base Address |
| 104h–107h | P0CLBU | Port 0 Command List Base Address Upper 32-Bits |
| 108h–10Bh | P0FB | Port 0 FIS Base Address |
| 10Ch–10Fh | P0FBU | Port 0 FIS Base Address Upper 32-Bits |
| 110h–113h | P0IS | Port 0 Interrupt Status |
| 114h–117h | P0IE | Port 0 Interrupt Enable |
| 118h–11Ch | P0CMD | Port 0 Command |
| 11Ch–11Fh | — | **Reserved** |
| 120h–123h | P0TFD | Port 0 Task File Data |
| 124h–127h | P0SIG | Port 0 Signature |
| 128h–12Bh | P0SSTS | Port 0 Serial ATA Status |
| 12Ch–12Fh | P0SCTL | Port 0 Serial ATA Control |
| 130h–133h | P0SERR | Port 0 Serial ATA Error |
| 134h–137h | P0SACT | Port 0 Serial ATA Active |
| 138h–13Bh | P0CI | Port 0 Command Issue |
| 13Ch–17Fh | — | **Reserved** |
| 180h–1FFh (Netbook Only) | — | **Reserved**<br>Registers are not available and software must not read from or write to registers. |

**Table 15-139.Port [1:0] DMA Register Address Map (Sheet 2 of 2)**

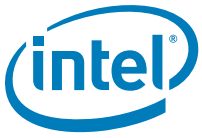
| ABAR + Offset | Mnemonic | Register |
|---|---|---|
| 180h–183h | P1CLB | Port 1 Command List Base Address |
| 184h–187h | P1CLBU | Port 1 Command List Base Address Upper 32-Bits |
| 188h–18Bh | P1FB | Port 1 FIS Base Address |
| 18Ch–18Fh | P1FBU | Port 1 FIS Base Address Upper 32-Bits |
| 190h–193h | P1IS | Port 1 Interrupt Status |
| 194h–197h | P1IE | Port 1 Interrupt Enable |
| 198h–19Ch | P1CMD | Port 1 Command |
| 19Ch–19Fh | — | **Reserved** |
| 1A0h–1A3h | P1TFD | Port 1 Task File Data |
| 1A4h–1A7h | P1SIG | Port 1 Signature |
| 1A8h–1ABh | P1SSTS | Port 1 Serial ATA Status |
| 1ACh–1AFh | P1SCTL | Port 1 Serial ATA Control |
| 1B0h–1B3h | P1SERR | Port 1 Serial ATA Error |
| 1B4h–1B7h | P1SACT | Port 1 Serial ATA Active |
| 1B8h–1BBh | P1CI | Port 1 Command Issue |
| 1BCh–1FFh | — | **Reserved** |
| 200h–2FFh | — | **Reserved** |

### 15.3.2.1 PxCLB—Port [1:0] Command List Base Address Register (D31:F2)

Address Offset: Port 0: ABAR + 100h<br>Port 1: ABAR + 180h — Attribute: R/W, RO

Default Value: Undefined — Size: 32 bits

| Bit | Description |
|---|---|
| 31:10 | **Command List Base Address (CLB)** — R/W**.** Indicates the 32-bit base for the command list for this port. This base is used when fetching commands to execute. The structure pointed to by this address range is 1 KB in length. This address must be 1-KB aligned as indicated by bits 31:10 being read/write.<br>Note that these bits are not reset on a HBA reset. |
| 9:0 | **Reserved** — RO |



### 15.3.2.2 PxCLBU—Port [1:0] Command List Base Address Upper 32-Bits Register (D31:F2)

Address Offset: Port 0: ABAR + 104h
Port 1: ABAR + 184h
Attribute: R/W

Default Value: Undefined
Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Command List Base Address Upper (CLBU)** — R/W**.** Indicates the upper 32-bits for the command list base address for this port. This base is used when fetching commands to execute.<br>Note that these bits are not reset on a HBA reset. |

### 15.3.2.3 PxFB—Port [1:0] FIS Base Address Register (D31:F2)

Address Offset: Port 0: ABAR + 108h
Port 1: ABAR + 188h
Attribute: R/W, RO

Default Value: Undefined
Size: 32 bits

| Bit | Description |
|---|---|
| 31:8 | **FIS Base Address (FB)** — R/W**.** Indicates the 32-bit base for received FISes. The structure pointed to by this address range is 256 bytes in length. This address must be 256-byte aligned, as indicated by bits 31:3 being read/write.<br>Note that these bits are not reset on a HBA reset. |
| 7:0 | **Reserved** — RO |

### 15.3.2.4 PxFBU—Port [1:0] FIS Base Address Upper 32-Bits Register (D31:F2)

Address Offset: Port 0: ABAR + 10Ch
Port 1: ABAR + 18Ch
Attribute: R/W

Default Value: Undefined
Size: 32 bits

| Bit | Description |
|---|---|
| 31:3 | **FIS Base Address Upper (FBU)** — R/W**.** Indicates the upper 32-bits for the received FIS base for this port.<br>Note that these bits are not reset on a HBA reset. |
| 2:0 | **Reserved** |

### 15.3.2.5 PxIS—Port [1:0] Interrupt Status Register (D31:F2)

Address Offset: Port 0: ABAR + 110h
Port 1: ABAR + 190h
Attribute: R/WC, RO
Default Value: 00000000h
Size: 32 bits

| Bit | Description |
|---|---|
| 31 | Cold Port Detect Status (CPDS) — RO. Cold presence not supported. |
| 30 | **Task File Error Status (TFES)** — R/WC. This bit is set whenever the status register is updated by the device and the error bit (PxTFD.bit 0) is set. |
| 29 | **Host Bus Fatal Error Status (HBFS)** — R/WC. Indicates that the Chipset encountered an error that it cannot recover from due to a bad software pointer. In PCI, such an indication would be a target or master abort. |
| 28 | **Host Bus Data Error Status (HBDS)** — R/WC. Indicates that the Chipset encountered a data error (uncorrectable ECC / parity) when reading from or writing to system memory. |
| 27 | **Interface Fatal Error Status (IFS)** — R/WC. Indicates that the Chipset encountered an error on the SATA interface which caused the transfer to stop. |
| 26 | **Interface Non-fatal Error Status (INFS)** — R/WC. Indicates that the Chipset encountered an error on the SATA interface but was able to continue operation. |
| 25 | **Reserved** |
| 24 | **Overflow Status (OFS)** — R/WC. Indicates that the Chipset received more bytes from a device than was specified in the PRD table for the command. |
| 23 | **Incorrect Port Multiplier Status (IPMS)** — R/WC. Indicates that the Chipset received a FIS from a device whose Port Multiplier field did not match what was expected.<br>**NOTE:** Port Multiplier not supported by Chipset. |
| 22 | PhyRdy Change Status (PRCS) — RO. When set to one indicates the internal PhyRdy signal changed state. This bit reflects the state of PxSERR.DIAG.N. Unlike most of the other bits in the register, this bit is RO and is only cleared when PxSERR.DIAG.N is cleared.<br>Note that the internal PhyRdy signal also transitions when the port interface enters partial or slumber power management states. Partial and slumber must be disabled when Surprise Removal Notification is desired, otherwise the power management state transitions will appear as false insertion and removal events. |
| 21:8 | **Reserved** |
| 7 | **Device Interlock Status (DIS)** — R/WC. When set, indicates that a platform interlock switch has been opened or closed, which may lead to a change in the connection state of the device.This bit is only valid in systems that support an interlock switch (CAP.SIS [ABAR+00:bit 28] set).<br>For systems that do not support an interlock switch, this bit will always be 0. |
| 6 | **Port Connect Change Status (PCS)** — RO. This bit reflects the state of PxSERR.DIAG.X. (ABAR+130h/1D0h/230h/2D0h, bit 26) Unlike other bits in this register, this bit is only cleared when PxSERR.DIAG.X is cleared.<br>0 = No change in Current Connect Status.<br>1 = Change in Current Connect Status. |
| 5 | **Descriptor Processed (DPS)** — R/WC. A PRD with the I bit set has transferred all its data. |

| Bit | Description |
|---|---|
| 4 | Unknown FIS Interrupt (UFS) — RO. When set to '1' indicates that an unknown FIS was received and has been copied into system memory. This bit is cleared to '0' by software clearing the PxSERR.DIAG.F bit to '0'. Note that this bit does not directly reflect the PxSERR.DIAG.F bit. PxSERR.DIAG.F is set immediately when an unknown FIS is detected, whereas this bit is set when the FIS is posted to memory. Software should wait to act on an unknown FIS until this bit is set to '1' or the two bits may become out of sync. |
| 3 | **Set Device Bits Interrupt (SDBS)** — R/WC**.** A Set Device Bits FIS has been received with the I bit set and has been copied into system memory. |
| 2 | **DMA Setup FIS Interrupt (DSS)** — R/WC**.** A DMA Setup FIS has been received with the I bit set and has been copied into system memory. |
| 1 | **PIO Setup FIS Interrupt (PSS)** — R/WC**.** A PIO Setup FIS has been received with the I bit set, it has been copied into system memory, and the data related to that FIS has been transferred. |
| 0 | **Device to Host Register FIS Interrupt (DHRS)** — R/WC**.** A D2H Register FIS has been received with the I bit set, and has been copied into system memory. |

### 15.3.2.6 PxIE—Port [1:0] Interrupt Enable Register (D31:F2)

Address Offset: Port 0: ABAR + 114h<br>Port 1: ABAR + 194h — Attribute: R/W, RO

Default Value: 00000000h — Size: 32 bits

This register enables and disables the reporting of the corresponding interrupt to system software. When a bit is set ('1') and the corresponding interrupt condition is active, then an interrupt is generated. Interrupt sources that are disabled ('0') are still reflected in the status registers.

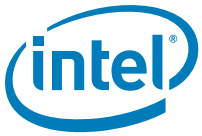
| Bit | Description |
|---|---|
| 31 | Cold Presence Detect Enable (CPDE) — RO. Cold Presence Detect not supported. |
| 30 | **Task File Error Enable (TFEE)** — R/W**.** When set, and GHC.IE and PxTFD.STS.ERR (due to a reception of the error register from a received FIS) are set, the Chipset will generate an interrupt. |
| 29 | **Host Bus Fatal Error Enable (HBFE)** — R/W**.** When set, and GHC.IE and PxS.HBFS are set, the Chipset will generate an interrupt. |
| 28 | **Host Bus Data Error Enable (HBDE)** — R/W**.** When set, and GHC.IE and PxS.HBDS are set, the Chipset will generate an interrupt. |
| 27 | **Host Bus Data Error Enable (HBDE)** — R/W. When set, GHC.IE is set, and PxIS.HBDS is set, the Chipset will generate an interrupt. |
| 26 | **Interface Non-fatal Error Enable (INFE)** — R/W. When set, GHC.IE is set, and PxIS.INFS is set, the Chipset will generate an interrupt. |
| 25 | **Reserved** - Should be written as 0 |
| 24 | **Overflow Error Enable (OFE)** — R/W**.** When set, and GHC.IE and PxS.OFS are set, the Chipset will generate an interrupt. |
| 23 | **Incorrect Port Multiplier Enable (IPME)** — R/W. When set, and GHC.IE and PxIS.IPMS are set, the Chipset will generate an interrupt.<br>**NOTE:** Should be written as 0. Port Multiplier not supported by Chipset. |

| Bit | Description |
|---|---|
| 22 | PhyRdy Change Interrupt Enable (PRCE) — R/W. When set, and GHC.IE is set, and PxIS.PRCS is set, the Chipset shall generate an interrupt. |
| 21:8 | **Reserved** - Should be written as 0 |
| 7 | **Device Interlock Enable (DIE)** — R/W. When set, and PxIS.DIS is set, the Chipset will generate an interrupt.<br>For systems that do not support an interlock switch, this bit shall be a read-only 0. |
| 6 | **Port Change Interrupt Enable (PCE)** — R/W**.** When set, and GHC.IE and PxS.PCS are set, the Chipset will generate an interrupt. |
| 5 | **Descriptor Processed Interrupt Enable (DPE)** — R/W**.** When set, and GHC.IE and PxS.DPS are set, the Chipset will generate an interrupt |
| 4 | **Unknown FIS Interrupt Enable (UFIE)** — R/W**.** When set, and GHC.IE is set and an unknown FIS is received, the Chipset will generate this interrupt. |
| 3 | **Set Device Bits FIS Interrupt Enable (SDBE)** — R/W**.** When set, and GHC.IE and PxS.SDBS are set, the Chipset will generate an interrupt. |
| 2 | **DMA Setup FIS Interrupt Enable (DSE)** — R/W**.** When set, and GHC.IE and PxS.DSS are set, the Chipset will generate an interrupt. |
| 1 | **PIO Setup FIS Interrupt Enable (PSE)** — R/W**.** When set, and GHC.IE and PxS.PSS are set, the Chipset will generate an interrupt. |
| 0 | **Device to Host Register FIS Interrupt Enable (DHRE)** — R/W**.** When set, and GHC.IE and PxS.DHRS are set, the Chipset will generate an interrupt. |

### 15.3.2.7 PxCMD—Port [1:0] Command Register (D31:F2)

Address Offset: Port 0: ABAR + 118h
Port 1: ABAR + 198h
Attribute: R/W, RO, R/WO

Default Value: 0000w00wh
where w = 00?0b (for?, see bit description)
Size: 32 bits



<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:28</td><td>Interface Communication Control (ICC) — R/W. This is a four bit field which can be used to control reset and power states of the interface. Writes to this field will cause actions on the interface, either as primitives or an OOB sequence, and the resulting status of the interface will be reported in the PxSSTS register (Address offset Port 0:ABAR+124h, Port 1: ABAR+1A4h, Port 2: ABAR+224h, Port 3: ABAR+2A4h).<br><br>
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>Fh–7h</td><td>Reserved</td></tr>
<tr><td>6h</td><td>Slumber: This will cause the Chipset to request a transition of the interface to the slumber state. The SATA device may reject the request and the interface will remain in its current state</td></tr>
<tr><td>5h–3h</td><td>Reserved</td></tr>
<tr><td>2h</td><td>Partial: This will cause the Chipset to request a transition of the interface to the partial state. The SATA device may reject the request and the interface will remain in its current state.</td></tr>
<tr><td>1h</td><td>Active: This will cause the Chipset to request a transition of the interface into the active</td></tr>
<tr><td>0h</td><td>No-Op / Idle: When software reads this value, it indicates the Chipset is not in the process of changing the interface state or sending a device reset, and a new link command may be issued.</td></tr>
</table>
When system software writes a non-reserved value other than No-Op (0h), the Chipset will perform the action and update this field back to Idle (0h).<br>
If software writes to this field to change the state to a state the link is already in (e.g. interface is in the active state and a request is made to go to the active state), the Chipset will take no action and return this field to Idle.<br>
NOTE: When the ALPE bit (bit 26) is set, then this register should not be set to 02h or 06h.</td></tr>
<tr><td>27</td><td>Aggressive Slumber / Partial (ASP) — R/W. When set, and the ALPE bit (bit 26) is set, the Chipset will aggressively enter the slumber state when it clears the PxCI register and the PxSACT register is cleared. When cleared, and the ALPE bit is set, the Chipset will aggressively enter the partial state when it clears the PxCI register and the PxSACT register is cleared.</td></tr>
<tr><td>26</td><td>Aggressive Link Power Management Enable (ALPE) — R/W. When set, the Chipset will aggressively enter a lower link power state (partial or slumber) based upon the setting of the ASP bit (bit 27).</td></tr>
<tr><td>25</td><td>Drive LED on ATAPI Enable (DLAE) — R/W. When set, the Chipset will drive the LED pin active for ATAPI commands (PxCLB[CHz.A] set) in addition to ATA commands. When cleared, the Chipset will only drive the LED pin active for ATA commands. See Section 5.17.5 - Volume 1 for details on the activity LED.</td></tr>
<tr><td>24</td><td>HDevice is ATAPI (ATAPI) — R/W. When set, the connected device is an ATAPI device. This bit is used by the Chipset to control whether or not to generate the Nettop LED when commands are active. See Section 5.17.5 - Volume 1 for details on the activity LED.</td></tr>
<tr><td>23:20</td><td>Reserved</td></tr>
</table>

| Bit | Description |
|---|---|
| 19 | **Interlock Switch Attached to Port (ISP)** — R/WO. When interlock switches are supported in the platform (CAP.SIS [ABAR+00h:bit 28] set), this indicates whether this particular port has an interlock switch attached. This bit can be used by system software to enable such features as aggressive power management, as disconnects can always be detected regardless of PHY state with an interlock switch. When this bit is set, it is expected that HPCP (bit 18) in this register is also set.<br>The Chipset takes no action on the state of this bit – it is for system software only. For example, if this bit is cleared, and an interlock switch toggles, the Chipset still treats it as a proper interlock switch event.<br>Note that these bits are not reset on a HBA reset. |
| 18 | **Hot Plug Capable Port (HPCP)** — R/WO.<br>0 = Port is not capable of Hot-Plug.<br>1 = Port is Hot-Plug capable.<br>This indicates whether the platform exposes this port to a device which can be Hot-Plugged. SATA by definition is hot-pluggable, but not all platforms are constructed to allow the device to be removed (it may be screwed into the chassis, for example). This bit can be used by system software to indicate a feature such as “eject device” to the end-user. The Chipset takes no action on the state of this bit - it is for system software only. For example, if this bit is cleared, and a Hot-Plug event occurs, the Chipset still treats it as a proper Hot-Plug event.<br>Note that these bits are not reset on a HBA reset. |
| 17 | **Port Multiplier Attached (PMA)** — RO / R/W. When this bit is set, a port multiplier is attached to the Chipset for this port. When cleared, a port multiplier is not attached to this port.<br>This bit is RO 0 when CAP.PMS (offset ABAR+00h:bit 17) = 0 and R/W when CAP.PMS = 1.<br>**NOTE:** Port Multiplier not supported by Chipset. |
| 16 | Port Multiplier FIS Based Switching Enable (PMFSE) — RO. The Chipset does not support FIS-based switching.<br>**NOTE:** Port Multiplier not supported by Chipset. |
| 15 | **Controller Running (CR)** — RO. When this bit is set, the DMA engines for a port are running. See section 5.2.2 of the *Serial ATA AHCI Specification* for details on when this bit is set and cleared by the Chipset. |
| 14 | **FIS Receive Running (FR)** — RO. When set, the FIS Receive DMA engine for the port is running. See section 12.2.2 of the *Serial ATA AHCI Specification* for details on when this bit is set and cleared by the Chipset. |
| 13 | **Interlock Switch State (ISS)** — RO. For systems that support interlock switches (via CAP.SIS [ABAR+00h:bit 28]), if an interlock switch exists on this port (via ISP in this register), this bit indicates the current state of the interlock switch. A 0 indicates the switch is closed, and a 1 indicates the switch is opened.<br>For systems that do not support interlock switches, or if an interlock switch is not attached to this port, this bit reports 0. |
| 12:8 | **Current Command Slot (CCS)** — RO**.** Indicates the current command slot the Chipset is processing. This field is valid when the ST bit is set in this register, and is constantly updated by the Chipset. This field can be updated as soon as the Chipset recognizes an active command slot, or at some point soon after when it begins processing the command.<br>This field is used by software to determine the current command issue location of the Chipset. In queued mode, software shall not use this field, as its value does not represent the current command being executed. Software shall only use PxCI and PxSACT when running queued commands. |
| 7:5 | **Reserved** |

| Bit | Description |
|---|---|
| 4 | **FIS Receive Enable (FRE)** — R/W. When set, the Chipset may post received FISes into the FIS receive area pointed to by PxFB (ABAR+108h/188h/208h/288h) and PxFBU (ABAR+10Ch/18Ch/20Ch/28Ch). When cleared, received FISes are not accepted by the Chipset, except for the first D2H (device-to-host) register FIS after the initialization sequence.<br>System software must not set this bit until PxFB (PxFBU) have been programmed with a valid pointer to the FIS receive area, and if software wishes to move the base, this bit must first be cleared, and software must wait for the FR bit (bit 14) in this register to be cleared. |
| 3 | Command List Override (CLO) — R/W. Setting this bit to '1' causes PxTFD.STS.BSY and PxTFD.STS.DRQ to be cleared to '0'. This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the PxTFD.STS register. The HBA sets this bit to '0' when PxTFD.STS.BSY and PxTFD.STS.DRQ have been cleared to '0'. A write to this register with a value of '0' shall have no effect.<br>This bit shall only be set to '1' immediately prior to setting the PxCMD.ST bit to '1' from a previous value of '0'. Setting this bit to '1' at any other time is not supported and will result in indeterminate behavior |
| 2 | Power On Device (POD) — RO**.** Cold presence detect not supported. Defaults to 1. |
| 1 | **Spin-Up Device (SUD)** — R/W / RO. This bit is R/W and defaults to 0 for systems that support staggered spin-up (R/W when CAP.SSS (ABAR+00h:bit 27) is 1). Bit is RO 1 for systems that do not support staggered spin-up (when CAP.SSS is 0).<br>0 = No action.<br>1 = On an edge detect from 0 to 1, the Chipset starts a COMRESET initialization sequence to the device. |
| 0 | **Start (ST)** — R/W**.** When set, the Chipset may process the command list. When cleared, the Chipset may not process the command list. Whenever this bit is changed from a 0 to a 1, the Chipset starts processing the command list at entry 0. Whenever this bit is changed from a 1 to a 0, the PxCI register is cleared by the Chipset upon the Chipset putting the controller into an idle state.<br>Refer to section 10.3.1 of the Serial ATA AHCI Specification for important restrictions on when ST can be set to 1. |

### 15.3.2.8 PxTFD—Port [1:0] Task File Data Register (D31:F2)

Address Offset: Port 0: ABAR + 120h<br>Port 1: ABAR + 1A0h — Attribute: RO

Default Value: 0000007Fh — Size: 32 bits

This is a 32-bit register that copies specific fields of the task file when FISes are received. The FISes that contain this information are:

D2H Register FIS
PIO Setup FIS
Set Device Bits FIS

| Bit | Description |
|---|---|
| 31:16 | **Reserved** |
| 15:8 | **Error (ERR)** — RO**.** Contains the latest copy of the task file error register. |
| 7:0 | **Status (STS)** — RO**.** Contains the latest copy of the task file status register. Fields of note in this register that affect AHCI.<br><br>Bit / Field / Definition:<br>7 / BSY / Indicates the interface is busy<br>6:4 / N/A / Not applicable<br>3 / DRQ / Indicates a data transfer is requested<br>2:1 / N/A / Not applicable<br>0 / ERR / Indicates an error during the transfer |

### 15.3.2.9 PxSIG—Port [1:0] Signature Register (D31:F2)

Address Offset: Port 0: ABAR + 124h<br>Port 1: ABAR + 1A4h — Attribute: RO

Default Value: FFFFFFFFh — Size: 32 bits

This is a 32-bit register which contains the initial signature of an attached device when the first D2H Register FIS is received from that device. It is updated once after a reset sequence.

| Bit | Description |
|---|---|
| 31:0 | **Signature (SIG)** — RO**.** Contains the signature received from a device on the first D2H register FIS. The bit order is as follows:<br><br>Bit / Field:<br>31:24 / LBA High Register<br>23:16 / LBA Mid Register<br>15:8 / LBA Low Register<br>7:0 / Sector Count Register |

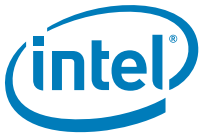
### 15.3.2.10 PxSSTS—Port [1:0] Serial ATA Status Register (D31:F2)

Address Offset: Port 0: ABAR + 128h<br>Port 1: ABAR + 1A8h — Attribute: RO

Default Value: 00000000h — Size: 32 bits

This is a 32-bit register that conveys the current state of the interface and host. The Chipset updates it continuously and asynchronously. When the Chipset transmits a COMRESET to the device, this register is updated to its reset values.

| Bit | Description |
|---|---|
| 31:12 | **Reserved** |



| Bit | Description |
|---|---|
| 11:8 | **Interface Power Management (IPM)** — RO. Indicates the current interface state:<br>Value / Description:<br>0h — Device not present or communication not established<br>1h — Interface in active state<br>2h — Interface in PARTIAL power management state<br>6h — Interface in SLUMBER power management state<br>All other values reserved. |
| 7:4 | Current Interface Speed (SPD) — RO. Indicates the negotiated interface communication speed.<br>Value / Description:<br>0h — Device not present or communication not established<br>1h — Generation 1 communication rate negotiated<br>2h — Generation 2 communication rate negotiated<br>All other values reserved.<br>Chipset supports Generation 1 communication rates (1.5 Gb/sec) and Gen 2 rates (3.0 Gb/s). |
| 3:0 | **Device Detection (DET)** — RO**.** Indicates the interface device detection and Phy state:<br>Value / Description:<br>0h — No device detected and Phy communication not established<br>1h — Device presence detected but Phy communication not established<br>3h — Device presence detected and Phy communication established<br>4h — Phy in offline mode as a result of the interface being disabled or running in a BIST loopback mode<br>All other values reserved. |

### 15.3.2.11 PxSCTL — Port [1:0] Serial ATA Control Register (D31:F2)

Address Offset: Port 0: ABAR + 12Ch<br>Port 1: ABAR + 1ACh — Attribute: R/W, RO

Default Value: 00000004h — Size: 32 bits

This is a 32-bit read-write register by which software controls SATA capabilities. Writes to the SControl register result in an action being taken by the Chipset or the interface. Reads from the register return the last value written to it.

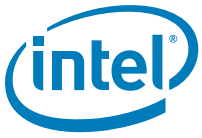
| Bit | Description |
|---|---|
| 31:20 | **Reserved** |
| 19:16 | Port Multiplier Port (PMP) — RO. This field is not used by AHCI<br>**NOTE:** Port Multiplier not supported by Chipset. |
| 15:12 | Select Power Management (SPM) — RO. This field is not used by AHCI |

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>11:8</td><td>Interface Power Management Transitions Allowed (IPM) — R/W. Indicates which power states the Chipset is allowed to transition to:<br><table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>No interface restrictions</td></tr><tr><td>1h</td><td>Transitions to the PARTIAL state disabled</td></tr><tr><td>2h</td><td>Transitions to the SLUMBER state disabled</td></tr><tr><td>3h</td><td>Transitions to both PARTIAL and SLUMBER states disabled</td></tr></table>All other values reserved</td></tr>
<tr><td>7:4</td><td>Speed Allowed (SPD) — R/W. Indicates the highest allowable speed of the interface. This speed is limited by the CAP.ISS (ABAR+00h:bit 23:20) field.<br><table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>No speed negotiation restrictions</td></tr><tr><td>1h</td><td>Limit speed negotiation to Generation 1 communication rate</td></tr><tr><td>2h</td><td>Limit speed negotiation to Generation 2 communication rate</td></tr></table>All other values reserved<br>Chipset Supports Generation 1 communication rates (1.5 Gb/sec) and Gen 2 rates (3.0 Gb/s).</td></tr>
<tr><td>3:0</td><td>Device Detection Initialization (DET) — R/W. Controls the chipset's device detection and interface initialization.<br><table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>No device detection or initialization action requested</td></tr><tr><td>1h</td><td>Perform interface communication initialization sequence to establish communication. This is functionally equivalent to a hard reset and results in the interface being reset and communications re-initialized</td></tr><tr><td>4h</td><td>Disable the Serial ATA interface and put Phy in offline mode</td></tr></table>All other values reserved.<br>When this field is written to a 1h, the Chipset initiates COMRESET and starts the initialization process. When the initialization is complete, this field shall remain 1h until set to another value by software.<br>This field may only be changed to 1h or 4h when PxCMD.ST is 0. Changing this field while the Chipset is running results in undefined behavior.</td></tr>
</table>

### 15.3.2.12 PxSERR—Port [1:0] Serial ATA Error Register (D31:F2)

Address Offset: Port 0: ABAR + 130h<br>Port 1: ABAR + 1B0h — Attribute: R/WC

Default Value: 00000000h — Size: 32 bits



| Bit | Description |
|---|---|
| 31:16 | **Diagnostics (DIAG)** — R/WC**.** Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes:<br>**Bits Description**<br>31:27**Reserved**<br>26 **Exchanged (X)**: When set to one this bit indicates a COMINIT signal was received. This bit is reflected in the interrupt register PxIS.PCS.<br>25 **Unrecognized FIS Type (F)**: Indicates that one or more FISs were received by the Transport layer with good CRC, but had a type field that was not recognized.<br>24 **Transport state transition error (T)**: Indicates that an error has occurred in the transition from one state to another within the Transport layer since the last time this bit was cleared.<br>23 **Link Sequence Error (S)**: Indicates that one or more Link state machine error conditions was encountered. The Link Layer state machine defines the conditions under which the link layer detects an erroneous transition.<br>22 **Handshake Error (H)**: Indicates that one or more R_ERR handshake response was received in response to frame transmission. Such errors may be the result of a CRC error detected by the recipient, a disparity or 8b/10b decoding error, or other error condition leading to a negative handshake on a transmitted frame.<br>21 **CRC Error (C)**: Indicates that one or more CRC errors occurred with the Link Layer.<br>20 **Disparity Error (D)**: This field is not used by AHCI.<br>19 **10b to 8b Decode Error (B)**: Indicates that one or more 10b to 8b decoding errors occurred.<br>18 **Comm Wake (W)**: Indicates that a Comm Wake signal was detected by the Phy.<br>17 **Phy Internal Error (I)**: Indicates that the Phy detected some internal error.<br>16 **PhyRdy Change (N)**: When set to 1 this bit indicates that the internal PhyRdy signal changed state since the last time this bit was cleared. In the Chipset, this bit will be set when PhyRdy changes from a 0 -> 1 or a 1 -> 0. The state of this bit is then reflected in the PxIS.PRCS interrupt status bit and an interrupt will be generated if enabled. Software clears this bit by writing a 1 to it. |
| 15:0 | **Error (ERR)** — R/WC**.** The ERR field contains error information for use by host software in determining the appropriate response to the error condition.<br>If one or more of bits 11:8 of this register are set, the controller will stop the current transfer.<br>**Bits Description**<br>15:12**Reserved**<br>11 **Interna**l Error (E): The SATA controller failed due to a master or target abort when attempting to access system memory.<br>10 Protocol Error (P): A violation of the Serial ATA protocol was detected. Note: The Chipset does not set this bit for all protocol violations that may occur on the SATA link.<br>9 Persistent Communication or Data Integrity Error (C): A communication error that was not recovered occurred that is expected to be persistent. Persistent communications errors may arise from faulty interconnect with the device, from a device that has been removed or has failed, or a number of other causes.<br>8 **Transient Data Integrity Error (T)**: A data integrity error occurred that was not recovered by the interface.<br>7:2 **Reserved**<br>1 **Recovered Communications Error (M)**: Communications between the device and host was temporarily lost but was re-established. This can arise from a device temporarily being removed, from a temporary loss of Phy synchronization, or from other causes and may be derived from the PhyNRdy signal between the Phy and Link layers.<br>0 **Recovered Data Integrity Error (I)**: A data integrity error occurred that was recovered by the interface through a retry operation or other recovery action. |

### 15.3.2.13 PxSACT—Port [1:0] Serial ATA Active (D31:F2)

Address Offset: Port 0: ABAR + 134h
Port 1: ABAR + 1B4h
Attribute: R/W

Default Value: 00000000h
Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Device Status (DS)** — R/W**.** System software sets this bit for SATA queuing operations prior to setting the PxCI.CI bit in the same command slot entry. This field is cleared via the Set Device Bits FIS.<br>This field is also cleared when PxCMD.ST (ABAR+118h/198h/218h/298h:bit 0) is cleared by software, and as a result of a COMRESET or SRST. |

### 15.3.2.14 PxCI—Port [1:0] Command Issue Register (D31:F2)

Address Offset: Port 0: ABAR + 138h
Port 1: ABAR + 1B8h
Attribute: R/W

Default Value: 00000000h
Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Commands Issued (CI)** — R/W**.** This field is set by software to indicate to the Chipset that a command has been built-in system memory for a command slot and may be sent to the device. When the Chipset receives a FIS which clears the BSY and DRQ bits for the command, it clears the corresponding bit in this register for that command slot.<br>This field is also cleared when PxCMD.ST (ABAR+118h/198h/218h/298h:bit 0) is cleared by software. |

§

# 16 EHCI Controller Registers (D29:F7)

## 16.1 USB EHCI Configuration Registers (USB EHCI—D29:F7)

***Note:*** Register address locations that are not shown in Table 16-140 should be treated as **Reserved** (see Section 9.2 for details).

***Note:*** All configuration registers in this section are in the core well and reset by a core well reset and the D3-to-D0 warm reset, except as noted.

**Table 16-140.USB EHCI PCI Register Address Map (USB EHCI—D29:F7) (Sheet 1 of 2)**

| Offset | Mnemonic | Register Name | Default Value | Type |
|---|---|---|---|---|
| 00h–01h | VID | Vendor Identification | 8086h | RO |
| 02h–03h | DID | Device Identification | See register description. | RO |
| 04h–05h | PCICMD | PCI Command | 0000h | R/W, RO |
| 06h–07h | PCISTS | PCI Status | 0290h | R/W, RO |
| 08h | RID | Revision Identification | See register description | RO |
| 09h | PI | Programming Interface | 20h | RO |
| 0Ah | SCC | Sub Class Code | 03h | RO |
| 0Bh | BCC | Base Class Code | 0Ch | RO |
| 0Dh | PMLT | Primary Master Latency Timer | 00h | RO |
| 10h–13h | MEM_BASE | Memory Base Address | 00000000h | R/W, RO |
| 2Ch–2Dh | SVID | USB EHCI Subsystem Vendor Identification | XXXXh | R/W (special) |
| 2Eh–2Fh | SID | USB EHCI Subsystem Identification | XXXXh | R/W (special) |
| 34h | CAP_PTR | Capabilities Pointer | 50h | RO |
| 3Ch | INT_LN | Interrupt Line | 00h | R/W |
| 3Dh | INT_PN | Interrupt Pin | See register description | RO |
| 50h | PWR_CAPID | PCI Power Management Capability ID | 01h | RO |
| 51h | NXT_PTR1 | Next Item Pointer #1 | 58h | R/W (special) |
| 52h–53h | PWR_CAP | Power Management Capabilities | C9C2h | R/W (special) |

**Table 16-140.USB EHCI PCI Register Address Map (USB EHCI—D29:F7) (Sheet 2 of 2)**

| Offset | Mnemonic | Register Name | Default Value | Type |
|---|---|---|---|---|
| 54h–55h | PWR_CNTL_STS | Power Management Control/Status | 0000h | R/W, R/WC, RO |
| 58h | DEBUG_CAPID | Debug Port Capability ID | 0Ah | RO |
| 59h | NXT_PTR2 | Next Item Pointer #2 | 00h | RO |
| 5Ah–5Bh | DEBUG_BASE | Debug Port Base Offset | 20A0h | RO |
| 60h | USB_RELNUM | USB Release Number | 20h | RO |
| 61h | FL_ADJ | Frame Length Adjustment | 20h | R/W |
| 62h–63h | PWAKE_CAP | Port Wake Capabilities | 01FFh | R/W |
| 64h–67h | — | **Reserved** | — | — |
| 68h–6Bh | LEG_EXT_CAP | USB EHCI Legacy Support Extended Capability | 00000001h | R/W, RO |
| 6Ch–6Fh | LEG_EXT_CS | USB EHCI Legacy Extended Support Control/Status | 00000000h | R/W, R/WC, RO |
| 70h–73h | SPECIAL_SMI | Intel Specific USB 2.0 SMI | 00000000h | R/W, R/WC |
| 74h–7Fh | — | **Reserved** | — | — |
| 80h | ACCESS_CNTL | Access Control | 00h | R/W |

## 16.1.1 VID—Vendor Identification Register (USB EHCI—D29:F7)

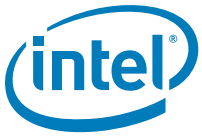
Offset Address: 00h–01h
Default Value: 8086h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Vendor ID** — RO. This is a 16-bit value assigned to Intel. |

## 16.1.2 DID—Device Identification Register (USB EHCI—D29:F7)

Offset Address: 02h–03h
Default Value: See bit description
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Device ID** — RO. This is a 16-bit value assigned to the Chipset USB EHCI controller. |



## 16.1.3 PCICMD—PCI Command Register (USB EHCI—D29:F7)

Address Offset: 0h4–05h Attribute: R/W, RO
Default Value: 0000h Size: 16 bits

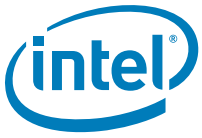
| Bit | Description |
|---|---|
| 15:11 | **Reserved** |
| 10 | **Interrupt Disable** — R/W.<br>0 = The function is capable of generating interrupts.<br>1 = The function can not generate its interrupt to the interrupt controller.<br>Note that the corresponding Interrupt Status bit (D29:F7:06h, bit 3) is not affected by the interrupt enable. |
| 9 | **Fast Back to Back Enable (FBE)** — RO. Hardwired to 0. |
| 8 | SERR# Enable (SERR_EN) — R/W.<br>0 = Disables EHC's capability to generate an SERR#.<br>1 = The Enhanced Host controller (EHC) is capable of generating (internally) SERR# when it receive a completion status other than "successful" for one of its DMA-initiated memory reads on DMI (and subsequently on its internal interface). |
| 7 | **Wait Cycle Control (WCC)** — RO. Hardwired to 0. |
| 6 | P**arity Error Response (PER)** — RO.<br>1 = EHCI Host Controller will check for correct parity and halt operation when bad parity is detected during the data phase as recommended by the EHCI specification. If it detects bad parity on the address or command phases when this bit is set to 1, the host controller does not take the cycle, halts the host controller (if currently not halted), and sets the host system error bit in the USBSTS register. Note that this applies to both requests and completions from the system interface.<br>This bit must be set in order for the parity errors to generate SERR#. |
| 5 | **VGA Palette Snoop (VPS)** — RO. Hardwired to 0. |
| 4 | **Postable Memory Write Enable (PMWE)** — RO. Hardwired to 0. |
| 3 | **Special Cycle Enable (SCE)** — RO. Hardwired to 0. |
| 2 | **Bus Master Enable (BME)** — R/W.<br>0 = Disables this functionality.<br>1 = Enables the Chipset to act as a master on the PCI bus for USB transfers. |
| 1 | **Memory Space Enable (MSE)** — R/W. This bit controls access to the USB 2.0 Memory Space registers.<br>0 = Disables this functionality.<br>1 = Enables accesses to the USB 2.0 registers. The Base Address register (D29:F7:10h) for USB 2.0 should be programmed before this bit is set. |
| 0 | **I/O Space Enable (IOSE)** — RO. Hardwired to 0. |

### 16.1.4 PCISTS—PCI Status Register (USB EHCI—D29:F7)

Address Offset: 06h–07h
Default Value: 0290h
Attribute: R/W, RO
Size: 16 bits

***Note:*** For the writable bits, software must write a 1 to clear bits that are set. Writing a 0 to the bit has no effect.

| Bit | Description |
|---|---|
| 15 | **Detected Parity Error (DPE)** — RO. Hardwired to 0. |
| 14 | **Signaled System Error (SSE)** — R/W.<br>0 = No SERR# signaled by Chipset.<br>1 = This bit is set by the Chipset when it signals SERR# (internally). The SER_EN bit (bit 8 of the Command Register) must be 1 for this bit to be set. |
| 13 | **Received Master Abort (RMA)** — R/W.<br>0 = No master abort received by EHC on a memory access.<br>1 = This bit is set when EHC, as a master, receives a master abort status on a memory access. This is treated as a Host Error and halts the DMA engines. This event can optionally generate an SERR# by setting the SERR# Enable bit. |
| 12 | **Received Target Abort (RTA)** — R/W.<br>0 = No target abort received by EHC on memory access.<br>1 = This bit is set when EHC, as a master, receives a target abort status on a memory access. This is treated as a Host Error and halts the DMA engines. This event can optionally generate an SERR# by setting the SERR# Enable bit (D29:F7:04h, bit 8). |
| 11 | **Signaled Target Abort (STA)** — RO. This bit is used to indicate when the EHCI function responds to a cycle with a target abort. There is no reason for this to happen, so this bit will be<br>hardwired to 0. |
| 10:9 | **DEVSEL# Timing Status (DEVT_STS)** — RO. This 2-bit field defines the timing for DEVSEL# assertion. |
| 8 | **Master Data Parity Error Detected (DPED)** — R/W.<br>0 = No data parity error detected on USB2.0 read completion packet.<br>1 = This bit is set by the Chipset when a data parity error is detected on a USB 2.0 read completion packet on the internal interface to the EHCI host controller and bit 6 of the Command register is set to 1. |
| 7 | **Fast Back to Back Capable (FB2BC)** — RO. Hardwired to 1. |
| 6 | **User Definable Features (UDF)** — RO. Hardwired to 0. |
| 5 | **66 MHz Capable (66 MHz _CAP)** — RO. Hardwired to 0. |
| 4 | **Capabilities List (CAP_LIST)** — RO. Hardwired to 1 indicating that offset 34h contains a valid capabilities pointer. |
| 3 | **Interrupt Status** — RO. This bit reflects the state of this function's interrupt at the input of the enable/disable logic.<br>0 = This bit will be 0 when the interrupt is de-asserted.<br>1 = This bit is a 1 when the interrupt is asserted.<br>The value reported in this bit is independent of the value in the Interrupt Enable bit. |
| 2:0 | **Reserved** |



### 16.1.5 RID—Revision Identification Register (USB EHCI—D29:F7)

Offset Address: 08h Attribute: RO
Default Value: See bit description Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Revision ID** — RO. |

### 16.1.6 PI—Programming Interface Register (USB EHCI—D29:F7)

Address Offset: 09h Attribute: RO
Default Value: 20h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Programming Interface** — RO. A value of 20h indicates that this USB 2.0 host controller conforms to the EHCI Specification. |

### 16.1.7 SCC—Sub Class Code Register (USB EHCI—D29:F7)

Address Offset: 0Ah Attribute: RO
Default Value: 03h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Sub Class Code (SCC)** — RO.<br>03h = Universal serial bus host controller. |

### 16.1.8 BCC—Base Class Code Register (USB EHCI—D29:F7)

Address Offset: 0Bh Attribute: RO
Default Value: 0Ch Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Base Class Code (BCC)** — RO.<br>0Ch = Serial bus controller. |

### 16.1.9 PMLT—Primary Master Latency Timer Register (USB EHCI—D29:F7)

Address Offset: 0Dh
Default Value: 00h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Master Latency Timer Count (MLTC)** — RO. Hardwired to 00h. Because the EHCI controller is internally implemented with arbitration on an interface (and not PCI), it does not need a master latency timer. |

### 16.1.10 MEM_BASE—Memory Base Address Register (USB EHCI—D29:F7)

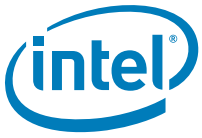
Address Offset: 10h–13h
Default Value: 00000000h
Attribute: R/W, RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:10 | **Base Address** — R/W. Bits [31:10] correspond to memory address signals [31:10], respectively. This gives 1-KB of locatable memory space aligned to 1-KB boundaries. |
| 9:4 | **Reserved** |
| 3 | **Prefetchable** — RO. Hardwired to 0 indicating that this range should not be prefetched. |
| 2:1 | **Type** — RO. Hardwired to 00b indicating that this range can be mapped anywhere within 32-bit address space. |
| 0 | **Resource Type Indicator (RTE)** — RO. Hardwired to 0 indicating that the base address field in this register maps to memory space. |

### 16.1.11 SVID—USB EHCI Subsystem Vendor ID Register (USB EHCI—D29:F7)

Address Offset: 2Ch–2Dh
Default Value: **XXXX**h
Reset: None
Attribute: R/W (special)
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Subsystem Vendor ID (SVID)** — R/W (special). This register, in combination with the USB 2.0 Subsystem ID register, enables the operating system to distinguish each subsystem from the others.<br><br>**NOTE:** Writes to this register are enabled when the WRT_RDONLY bit (D29:F7:80h, bit 0) is set to 1. |



### 16.1.12 SID—USB EHCI Subsystem ID Register (USB EHCI—D29:F7)

Address Offset: 2Eh–2Fh
Default Value: XXXXh
Reset: None
Attribute: R/W (special)
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Subsystem ID (SID)** — R/W (special). BIOS sets the value in this register to identify the Subsystem ID. This register, in combination with the Subsystem Vendor ID register, enables the operating system to distinguish each subsystem from other(s).<br><br>**NOTE:** Writes to this register are enabled when the WRT_RDONLY bit (D29:F7:80h, bit 0) is set to 1. |

### 16.1.13 CAP_PTR—Capabilities Pointer Register (USB EHCI—D29:F7)

Address Offset: 34h
Default Value: 50h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Capabilities Pointer (CAP_PTR)** — RO. This register points to the starting offset of the USB 2.0 capabilities ranges. |

### 16.1.14 INT_LN—Interrupt Line Register (USB EHCI—D29:F7)

Address Offset: 3Ch
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Interrupt Line (INT_LN)** — R/W. This data is not used by the Chipset. It is used as a scratchpad register to communicate to software the interrupt line that the interrupt pin is connected to. |

### 16.1.15 INT_PN—Interrupt Pin Register (USB EHCI—D29:F7)

Address Offset: 3Dh
Default Value: See Description
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Interrupt Pin** — RO. This reflects the value of D29IP.EIP (Chipset Config Registers:Offset 3108:bits 31:28).<br>**NOTE:** Bits 7:4 are always 0h |

### 16.1.16 PWR_CAPID—PCI Power Management Capability ID Register (USB EHCI—D29:F7)

Address Offset: 50h
Default Value: 01h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Power Management Capability ID** — RO. A value of 01h indicates that this is a PCI Power Management capabilities field. |

### 16.1.17 NXT_PTR1—Next Item Pointer #1 Register (USB EHCI—D29:F7)

Address Offset: 51h
Default Value: 58h
Attribute: R/W (special)
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Next Item Pointer 1 Value** — R/W (special). This register defaults to 58h, which indicates that the next capability registers begin at configuration offset 58h. This register is writable when the WRT_RDONLY bit (D29:F7:80h, bit 0) is set. This allows BIOS to effectively hide the Debug Port capability registers, if necessary. This register should only be written during system initialization before the plug-and-play software has enabled any master-initiated traffic. Only values of 58h (Debug Port visible) and 00h (Debug Port invisible) are expected to be programmed in this register.<br>**NOTE:** Register not reset by D3-to-D0 warm reset. |

### 16.1.18 PWR_CAP—Power Management Capabilities Register (USB EHCI—D29:F7)

Address Offset: 52h–53h
Default Value: C9C2h
Attribute: R/W (special)
Size: 16 bits

| Bit | Description |
|---|---|
| 15:11 | **PME Support (PME_SUP)** — R/W (special). This 5-bit field indicates the power states in which the function may assert PME#. The Chipset EHC does not support the D1 or D2 states. For all other states, the Chipset EHC is capable of generating PME#. Software should not need to modify this field. |
| 10 | **D2 Support (D2_SUP)** — R/W (special).<br>0 = D2 State is not supported<br>1 = D2 State is supported |
| 9 | **D1 Support (D1_SUP)** — R/W (special).<br>0 = D1 State is not supported<br>1 = D1 State is supported |
| 8:6 | **Auxiliary Current (AUX_CUR)** — R/W (special). The Chipset EHC reports 375 mA maximum suspend well current required when in the $D3_{COLD}$ state. This value can be written by BIOS when a more accurate value is known. |
| 5 | **Device Specific Initialization (DSI)**— R/W (special). The Chipset reports 0, indicating that no<br>device-specific initialization is required. |

| Bit | Description |
|---|---|
| 4 | **Reserved** |
| 3 | **PME Clock (PME_CLK)** — R/W (special). The Chipset reports 0, indicating that no PCI clock is required to generate PME#. |
| 2:0 | **Version (VER)** — R/W (special). The Chipset reports 010b, indicating that it complies with Revision 1.1 of the PCI Power Management Specification. |

**NOTES:**
1. Normally, this register is read-only to report capabilities to the power management software. To report different power management capabilities, depending on the system in which the Chipset is used, bits 15:11 and 8:6 in this register are writable when the WRT_RDONLY bit (D29:F7:80h, bit 0) is set. The value written to this register does not affect the hardware other than changing the value returned during a read.
2. Reset: core well, but not D3-to-D0 warm reset.

## 16.1.19 PWR_CNTL_STS—Power Management Control/ Status Register (USB EHCI—D29:F7)

Address Offset: 54h–55h Attribute: R/W, R/WC, RO
Default Value: 0000h Size: 16 bits

| Bit | Description |
|---|---|
| 15 | **PME Status** — R/WC.<br>0 = Writing a 1 to this bit will clear it and cause the internal PME to deassert (if enabled).<br>1 = This bit is set when the Chipset EHC would normally assert the PME# signal independent of the state of the PME_En bit.<br><br>**NOTE:** This bit must be explicitly cleared by the operating system each time the operating system is loaded. |
| 14:13 | **Data Scale** — RO. Hardwired to 00b indicating it does not support the associated Data register. |
| 12:9 | **Data Select** — RO. Hardwired to 0000b indicating it does not support the associated Data register. |
| 8 | **PME Enable** — R/W.<br>0 = Disable.<br>1 = Enable. Enables Chipset EHC to generate an internal PME signal when PME_Status is 1.<br><br>**NOTE:** This bit must be explicitly cleared by the operating system each time it is initially loaded. |
| 7:2 | **Reserved** |

| Bit | Description |
|---|---|
| 1:0 | **Power State** — R/W. This 2-bit field is used both to determine the current power state of EHC function and to set a new power state. The definition of the field values are:<br>00 = D0 state<br>11 = $D3_{HOT}$ state<br>If software attempts to write a value of 10b or 01b in to this field, the write operation must complete normally; however, the data is discarded and no state change occurs. When in the $D3_{HOT}$ state, the Chipset must not accept accesses to the EHC memory range; but the configuration space must still be accessible. When not in the D0 state, the generation of the interrupt output is blocked. Specifically, the PIRQH is not asserted by the Chipset when not in the D0 state.<br>When software changes this value from the $D3_{HOT}$ state to the D0 state, an internal warm (soft) reset is generated, and software must re-initialize the function. |

**NOTE:** Reset (bits 15, 8): suspend well, and not D3-to-D0 warm reset nor core well reset.

## 16.1.20 DEBUG_CAPID—Debug Port Capability ID Register (USB EHCI—D29:F7)

Address Offset: 58h
Default Value: 0Ah
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Debug Port Capability ID** — RO. Hardwired to 0Ah indicating that this is the start of a Debug Port Capability structure. |

## 16.1.21 NXT_PTR2—Next Item Pointer #2 Register (USB EHCI—D29:F7)

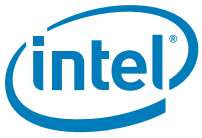
Address Offset: 59h
Default Value: 00h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Next Item Pointer 2 Capability** — RO. Hardwired to 00h to indicate there are no more capability structures in this function. |

## 16.1.22 DEBUG_BASE—Debug Port Base Offset Register (USB EHCI—D29:F7)

Address Offset: 5Ah–5Bh
Default Value: 20A0h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:13 | **BAR Number** — RO. Hardwired to 001b to indicate the memory BAR begins at offset 10h in the EHCI configuration space. |
| 12:0 | **Debug Port Offset** — RO. Hardwired to 0A0h to indicate that the Debug Port registers begin at offset A0h in the EHCI memory range. |



### 16.1.23 USB_RELNUM—USB Release Number Register (USB EHCI—D29:F7)

Address Offset: 60h
Default Value: 20h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **USB Release Number** — RO. A value of 20h indicates that this controller follows *Universal Serial Bus (USB) Specification, Revision 2.0*. |

### 16.1.24 FL_ADJ—Frame Length Adjustment Register (USB EHCI—D29:F7)

Address Offset: 61h
Default Value: 20h
Attribute: R/W
Size: 8 bits

This feature is used to adjust any offset from the clock source that generates the clock that drives the SOF counter. When a new value is written into these six bits, the length of the frame is adjusted. Its initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. This register should only be modified when the HChalted bit (D29:F7:CAPLENGTH + 24h, bit 12) in the USB2.0_STS register is a 1. Changing value of this register while the host controller is operating yields undefined results. It should not be reprogrammed by USB system software unless the default or BIOS programmed values are incorrect, or the system is restoring the register while returning from a suspended state.

These bits in suspend well and not reset by a D3-to-D0 warm rest or a core well reset.

| Bit | Description |
|---|---|
| 7:6 | **Reserved** — RO. These bits are reserved for future use and should read as 00b. |
| 5:0 | **Frame Length Timing Value** — R/W. Each decimal value change to this register corresponds to 16 high-speed bit times. The SOF cycle time (number of SOF counter clock periods to generate a SOF micro-frame length) is equal to 59488 + value in this field. The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. |

| Frame Length (# 480 MHz Clocks) (decimal) | Frame Length Timing Value (this register) (decimal) |
|---|---|
| 59488 | 0 |
| 59504 | 1 |
| 59520 | 2 |
| — | — |
| 59984 | 31 |
| 60000 | 32 |
| — | — |
| 60480 | 62 |
| 60496 | 63 |

### 16.1.25 PWAKE_CAP—Port Wake Capability Register (USB EHCI—D29:F7)

Address Offset: 62–63h
Default Value: 01FFh
Attribute: R/W
Size: 16 bits

This register is in the suspend power well. The intended use of this register is to establish a policy about which ports are to be used for wake events. Bit positions 1–8 in the mask correspond to a physical port implemented on the current EHCI controller. A 1 in a bit position indicates that a device connected below the port can be enabled as a wake-up device and the port may be enabled for disconnect/connect or overcurrent events as wake-up events. This is an information-only mask register. The bits in this register **do not** affect the actual operation of the EHCI host controller. The system-specific policy can be established by BIOS initializing this register to a system-specific value. System software uses the information in this register when enabling devices and ports for remote wake-up.

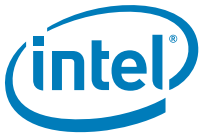
These bits are not reset by a D3-to-D0 warm rest or a core well reset.

| Bit | Description |
|---|---|
| 15:9 | **Reserved** — RO. |
| 8:1 | **Port Wake Up Capability Mask** — R/W. Bit positions 1 through 8 correspond to a physical port implemented on this host controller. For example, bit position 1 corresponds to port 0, bit position 2 corresponds to port 1, etc. |
| 0 | **Port Wake Implemented** — R/W. A 1 in this bit indicates that this register is implemented to software. |

### 16.1.26 LEG_EXT_CAP—USB EHCI Legacy Support Extended Capability Register (USB EHCI—D29:F7)

Address Offset: 68–6Bh
Default Value: 00000001h
Attribute: R/W, RO
Size: 32 bits
Power Well: Suspend
**NOTE:** These bits are not reset by a D3-to-D0 warm rest or a core well reset.

| Bit | Description |
|---|---|
| 31:25 | **Reserved** — RO. Hardwired to 00h |
| 24 | **HC OS Owned Semaphore** — R/W. System software sets this bit to request ownership of the EHCI controller. Ownership is obtained when this bit reads as 1 and the HC BIOS Owned Semaphore bit reads as clear. |
| 23:17 | **Reserved** — RO. Hardwired to 00h |
| 16 | **HC BIOS Owned Semaphore** — R/W. The BIOS sets this bit to establish ownership of the EHCI controller. System BIOS will clear this bit in response to a request for ownership of the EHCI controller by system software. |
| 15:8 | **Next EHCI Capability Pointer** — RO. Hardwired to 00h to indicate that there are no EHCI Extended Capability structures in this device. |
| 7:0 | **Capability ID** — RO. Hardwired to 01h to indicate that this EHCI Extended Capability is the Legacy Support Capability. |



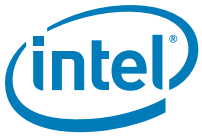
## 16.1.27 LEG_EXT_CS—USB EHCI Legacy Support Extended Control / Status Register (USB EHCI—D29:F7)

Address Offset: 6C–6Fh
Attribute: R/W, R/WC, RO
Default Value: 00000000h
Size: 32 bits
Power Well: Suspend

**NOTE:** These bits are not reset by a D3-to-D0 warm rest or a core well reset.

| Bit | Description |
|---|---|
| 31 | **SMI on BAR** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = Base Address Register (BAR) not written.<br>1 = This bit is set to 1 when the Base Address Register (BAR) is written. |
| 30 | **SMI on PCI Command** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = PCI Command (PCICMD) Register Not written.<br>1 = This bit is set to 1 when the PCI Command (PCICMD) Register is written. |
| 29 | **SMI on OS Ownership Change** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = No HC OS Owned Semaphore bit change.<br>1 = This bit is set to 1 when the HC OS Owned Semaphore bit in the LEG_EXT_CAP register (D29:F7:68h, bit 24) transitions from 1 to 0 or 0 to 1. |
| 28:22 | **Reserved** — RO. Hardwired to 00h |
| 21 | **SMI on Async Advance** — RO. This bit is a shadow bit of the Interrupt on Async Advance bit (D29:F7:CAPLENGTH + 24h, bit 5) in the USB2.0_STS register.<br><br>**NOTE:** To clear this bit system software must write a 1 to the Interrupt on Async Advance bit in the USB2.0_STS register. |
| 20 | **SMI on Host System Error** — RO. This bit is a shadow bit of Host System Error bit in the USB2.0_STS register (D29:F7:CAPLENGTH + 24h, bit 4).<br><br>**NOTE:** To clear this bit system software must write a 1 to the Host System Error bit in the USB2.0_STS register. |
| 19 | **SMI on Frame List Rollover** — RO. This bit is a shadow bit of Frame List Rollover bit (D29:F7:CAPLENGTH + 24h, bit 3) in the USB2.0_STS register.<br><br>**NOTE:** To clear this bit system software must write a 1 to the Frame List Rollover bit in the USB2.0_STS register. |
| 18 | **SMI on Port Change Detect** — RO. This bit is a shadow bit of Port Change Detect bit (D29:F7:CAPLENGTH + 24h, bit 2) in the USB2.0_STS register.<br><br>**NOTE:** To clear this bit system software must write a 1 to the Port Change Detect bit in the USB2.0_STS register. |
| 17 | **SMI on USB Error** — RO. This bit is a shadow bit of USB Error Interrupt (USBERRINT) bit (D29:F7:CAPLENGTH + 24h, bit 1) in the USB2.0_STS register.<br><br>**NOTE:** To clear this bit system software must write a 1 to the USB Error Interrupt bit in the USB2.0_STS register. |
| 16 | **SMI on USB Complete** — RO. This bit is a shadow bit of USB Interrupt (USBINT) bit (D29:F7:CAPLENGTH + 24h, bit 0) in the USB2.0_STS register.<br><br>**NOTE:** To clear this bit system software must write a 1 to the USB Interrupt bit in the USB2.0_STS register. |

| Bit | Description |
|---|---|
| 15 | **SMI on BAR Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is 1 and SMI on BAR (D29:F7:6Ch, bit 31) is 1, then the host controller will issue an SMI. |
| 14 | **SMI on PCI Command Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is 1 and SMI on PCI Command (D29:F7:6Ch, bit 30) is 1, then the host controller will issue an SMI. |
| 13 | **SMI on OS Ownership Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1 AND the OS Ownership Change bit (D29:F7:6Ch, bit 29) is 1, the host controller will issue an SMI. |
| 12:6 | **Reserved** — RO. Hardwired to 00h |
| 5 | **SMI on Async Advance Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1, and the SMI on Async Advance bit (D29:F7:6Ch, bit 21) is a 1, the host controller will issue an SMI immediately. |
| 4 | **SMI on Host System Error Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1, and the SMI on Host System Error (D29:F7:6Ch, bit 20) is a 1, the host controller will issue an SMI. |
| 3 | **SMI on Frame List Rollover Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1, and the SMI on Frame List Rollover bit (D29:F7:6Ch, bit 19) is a 1, the host controller will issue an SMI. |
| 2 | **SMI on Port Change Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1, and the SMI on Port Change Detect bit (D29:F7:6Ch, bit 18) is a 1, the host controller will issue an SMI. |
| 1 | **SMI on USB Error Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1, and the SMI on USB Error bit (D29:F7:6Ch, bit 17) is a 1, the host controller will issue an SMI immediately. |
| 0 | **SMI on USB Complete Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1, and the SMI on USB Complete bit (D29:F7:6Ch, bit 16) is a 1, the host controller will issue an SMI immediately. |



## 16.1.28 SPECIAL_SMI—Intel Specific USB 2.0 SMI Register (USB EHCI—D29:F7)

Address Offset: 70h–73h Attribute: R/W, R/WC
Default Value: 00000000h Size: 32 bits
Power Well: Suspend
**NOTE:** These bits are not reset by a D3-to-D0 warm rest or a core well reset.

| Bit | Description |
|---|---|
| 31:30 | **Reserved** — RO. Hardwired to 00h |
| 29:22 | **SMI on PortOwner** — R/WC. Software clears these bits by writing a 1 to it.<br>0 = No Port Owner bit change.<br>1 = Bits 29:22 correspond to the Port Owner bits for ports 1 (22) through 8 (29). These bits are set to 1 when the associated Port Owner bits transition from 0 to 1 or 1 to 0. |
| 21 | **SMI on PMCSR** — R/WC. Software clears these bits by writing a 1 to it.<br>0 = Power State bits Not modified.<br>1 = Software modified the Power State bits in the Power Management Control/Status (PMCSR) register (D29:F7:54h). |
| 20 | **SMI on Async** — R/WC. Software clears these bits by writing a 1 to it.<br>0 = No Async Schedule Enable bit change<br>1 = Async Schedule Enable bit transitioned from 1 to 0 or 0 to 1. |
| 19 | **SMI on Periodic** — R/WC. Software clears this bit by writing a 1 it.<br>0 = No Periodic Schedule Enable bit change.<br>1 = Periodic Schedule Enable bit transitions from 1 to 0 or 0 to 1. |
| 18 | **SMI on CF** — R/WC. Software clears this bit by writing a 1 it.<br>0 = No Configure Flag (CF) change.<br>1 = Configure Flag (CF) transitions from 1 to 0 or 0 to 1. |
| 17 | **SMI on HCHalted** — R/WC. Software clears this bit by writing a 1 it.<br>0 = HCHalted did Not transition to 1 (as a result of the Run/Stop bit being cleared).<br>1 = HCHalted transitions to 1 (as a result of the Run/Stop bit being cleared). |
| 16 | **SMI on HCReset** — R/WC. Software clears this bit by writing a 1 it.<br>0 = HCRESET did Not transitioned to 1.<br>1 = HCRESET transitioned to 1. |
| 15:14 | **Reserved** — RO. Hardwired to 00h |
| 13:6 | **SMI on PortOwner Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When any of these bits are 1 and the corresponding SMI on PortOwner bits are 1, then the host controller will issue an SMI. Unused ports should have their corresponding bits cleared. |
| 5 | **SMI on PMSCR Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is 1 and SMI on PMSCR is 1, then the host controller will issue an SMI. |
| 4 | **SMI on Async Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is 1 and SMI on Async is 1, then the host controller will issue an SMI |

| Bit | Description |
|---|---|
| 3 | **SMI on Periodic Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is 1 and SMI on Periodic is 1, then the host controller will issue an SMI. |
| 2 | **SMI on CF Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is 1 and SMI on CF is 1, then the host controller will issue an SMI. |
| 1 | **SMI on HCHalted Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1 and SMI on HCHalted is 1, then the host controller will issue an SMI. |
| 0 | **SMI on HCReset Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1 and SMI on HCReset is 1, then host controller will issue an SMI. |

### 16.1.29 ACCESS_CNTL—Access Control Register (USB EHCI—D29:F7)

Address Offset: 80h Attribute: R/W
Default Value: 00h Size: 8 bits

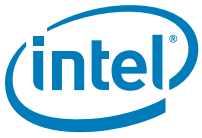
| Bit | Description |
|---|---|
| 7:1 | **Reserved** |
| 0 | **WRT_RDONLY** — R/W. When set to 1, this bit enables a select group of normally read-only registers in the EHC function to be written by software. Registers that may only be written when this mode is entered are noted in the summary tables and detailed description as “Read/Write-Special”. The registers fall into two categories:<br>1. System-configured parameters, and<br>2. Status bits |

## 16.2 Memory-Mapped I/O Registers

The EHCI memory-mapped I/O space is composed of two sets of registers: Capability Registers and Operational Registers.

***Note:*** The Chipset EHCI controller will not accept memory transactions (neither reads nor writes) as a target that are locked transactions. The locked transactions should not be forwarded to PCI as the address space is known to be allocated to USB.

***Note:*** When the EHCI function is in the D3 PCI power state, accesses to the USB 2.0 memory range are ignored and result a master abort. Similarly, if the Memory Space Enable (MSE) bit (D29:F7:04h, bit 1) is not set in the Command register in configuration space, the memory range will not be decoded by the Chipset enhanced host controller (EHC). If the MSE bit is not set, then the Chipset must default to allowing any memory accesses for the range specified in the BAR to go to PCI. This is because the range may



not be valid and, therefore, the cycle must be made available to any other targets that may be currently using that range.

## 16.2.1 Host Controller Capability Registers

These registers specify the limits, restrictions and capabilities of the host controller implementation. Within the host controller capability registers, only the structural parameters register is writable. These registers are implemented in the suspend well and is only reset by the standard suspend-well hardware reset, not by HCRESET or the D3-to-D0 reset.

**Table 16-141.Enhanced Host Controller Capability Registers**

| MEM_BASE + Offset | Mnemonic | Register | Default | Type |
|---|---|---|---|---|
| 00h | CAPLENGTH | Capabilities Registers Length | 20h | RO |
| 02h–03h | HCIVERSION | Host Controller Interface Version Number | 0100h | RO |
| 04h–07h | HCSPARAMS | Host Controller Structural Parameters | 00104208h | R/W (special), RO |
| 08h–0Bh | HCCPARAMS | Host Controller Capability Parameters | 00006871h | RO |

**NOTE:** "Read/Write Special" means that the register is normally read-only, but may be written when the WRT_RDONLY bit is set. Because these registers are expected to be programmed by BIOS during initialization, their contents must not get modified by HCRESET or D3-to-D0 internal reset.

### 16.2.1.1 CAPLENGTH—Capability Registers Length Register

Offset: MEM_BASE + 00h Attribute: RO
Default Value: 20h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Capability Register Length Value** — RO. This register is used as an offset to add to the Memory Base Register (D29:F7:10h) to find the beginning of the Operational Register Space. This field is hardwired to 20h indicating that the Operation Registers begin at offset 20h. |

### 16.2.1.2 HCIVERSION—Host Controller Interface Version Number Register

Offset: MEM_BASE + 02h–03h Attribute: RO
Default Value: 0100h Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Host Controller Interface Version Number** — RO. This is a two-byte register containing a BCD encoding of the version number of interface that this host controller interface conforms. |

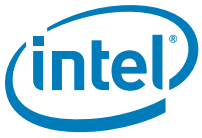
### 16.2.1.3 HCSPARAMS—Host Controller Structural Parameters

| | | | |
|---|---|---|---|
| Offset: | MEM_BASE + 04h–07h | Attribute: | R/W (special), RO |
| Default Value: | 00104208h | Size: | 32 bits |

***Note:*** This register is reset by a suspend well reset and not a D3-to-D0 reset or HCRESET.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** — RO. Default=0h. |
| 23:20 | **Debug Port Number (DP_N)** — RO (special). Hardwired to 1h indicating that the Debug Port is on the lowest numbered port on the Chipset. |
| 19:16 | **Reserved** |
| 15:12 | **Number of Companion Controllers (N_CC)** — R/W (special). This field indicates the number of companion controllers associated with this USB EHCI host controller.<br>A 0 in this field indicates there are no companion host controllers. Port-ownership hand-off is not supported. Only high-speed devices are supported on the host controller root ports.<br>A value of 1 or more in this field indicates there are companion USB UHCI host controller(s). Port-ownership hand-offs are supported. High, Full- and Low-speed devices are supported on the host controller root ports.<br>The Chipset allows the default value of 4h to be over-written by BIOS. When removing classic controllers, they should be disabled in the following order: Function 3, Function 2, Function 1, and Function 0, which correspond to ports 7:6, 5:4, 3:2, and 1:0, respectively. |
| 11:8 | **Number of Ports per Companion Controller (N_PCC)** — RO. Hardwired to 2h. This field indicates the number of ports supported per companion host controller. It is used to indicate the port routing configuration to system software. |
| 7:4 | **Reserved**. These bits are reserved and default to 0. |
| 3:0 | **N_PORTS** — R/W (special). This field specifies the number of physical downstream ports implemented on this host controller. The value of this field determines how many port registers are addressable in the Operational Register Space. Valid values are in the range of 1h to Fh.<br>The Chipset reports 8h by default. However, software may write a value less than 8 for some platform configurations. A 0 in this field is undefined. |

**NOTE:** This register is writable when the WRT_RDONLY bit is set.



#### 16.2.1.4 HCCPARAMS—Host Controller Capability Parameters Register

Offset: MEM_BASE + 08h–0Bh Attribute: RO
Default Value: 00006871h Size: 32 bits

| Bit | Description |
|---|---|
| 31:16 | **Reserved** |
| 15:8 | **EHCI Extended Capabilities Pointer (EECP)** — RO. This field is hardwired to 68h, indicating that the EHCI capabilities list exists and begins at offset 68h in the PCI configuration space. |
| 7:4 | **Isochronous Scheduling Threshold** — RO. This field indicates, relative to the current position of the executing host controller, where software can reliably update the isochronous schedule. When bit 7 is 0, the value of the least significant 3 bits indicates the number of micro-frames a host controller hold a set of isochronous data structures (one or more) before flushing the state. When bit 7 is a 1, then host software assumes the host controller may cache an isochronous data structure for an entire frame. Refer to the EHCI specification for details on how software uses this information for scheduling isochronous transfers.<br>This field is hardwired to 7h. |
| 3 | **Reserved**. These bits are reserved and should be set to 0. |
| 2 | **Asynchronous Schedule Park Capability** — RO. This bit is hardwired to 0 indicating that the host controller does not support this optional feature |
| 1 | **Programmable Frame List Flag** — RO.<br>0 = System software must use a frame list length of 1024 elements with this host controller. The USB2.0_CMD register (D29:F7:CAPLENGTH + 20h, bits 3:2) *Frame List Size* field is a read-only register and must be set to 0.<br>1 = System software can specify and use a smaller frame list and configure the host controller via the USB2.0_CMD register *Frame List Size* field. The frame list must always be aligned on a 4K page boundary. This requirement ensures that the frame list is always physically contiguous. |
| 0 | **64-bit Addressing Capability** — RO. This field documents the addressing range capability of this implementation. The value of this field determines whether software should use the 32-bit or 64-bit data structures. Values for this field have the following interpretation:<br>0 = Data structures using 32-bit address memory pointers<br>1 = Data structures using 64-bit address memory pointers<br>This bit is hardwired to 1.<br>**NOTE:** Chipset only implements 44 bits of addressing. Bits 63:44 will always be 0. |

### 16.2.2 Host Controller Operational Registers

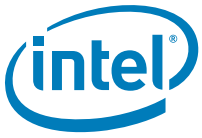
This section defines the enhanced host controller operational registers. These registers are located after the capabilities registers. The operational register base must be DWord-aligned and is calculated by adding the value in the first capabilities register (CAPLENGTH) to the base address of the enhanced host controller register address space (MEM_BASE). Since CAPLENGTH is always 20h, Table 16-142 already accounts for this offset. All registers are 32 bits in length.

### Table 16-142.Enhanced Host Controller Operational Register Address Map

| MEM_BASE + Offset | Mnemonic | Register Name | Default | Special Notes | Type |
|---|---|---|---|---|---|
| 20h–23h | USB2.0_CMD | USB 2.0 Command | 00080000h | | R/W, RO |
| 24h–27h | USB2.0_STS | USB 2.0 Status | 00001000h | | R/WC, RO |
| 28h–2Bh | USB2.0_INTR | USB 2.0 Interrupt Enable | 00000000h | | R/W |
| 2Ch–2Fh | FRINDEX | USB 2.0 Frame Index | 00000000h | | R/W, |
| 30h–33h | CTRLDSSEGMENT | Control Data Structure Segment | 00000000h | | R/W, RO |
| 34h–37h | PERODICLISTBASE | Period Frame List Base Address | 00000000h | | R/W |
| 38h–3Bh | ASYNCLISTADDR | Current Asynchronous List Address | 00000000h | | R/W |
| 3Ch–5Fh | — | **Reserved** | 0h | | RO |
| 60h–63h | CONFIGFLAG | Configure Flag | 00000000h | Suspend | R/W |
| 64h–67h | PORT0SC | Port 0 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 68h–6Bh | PORT1SC | Port 1 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 6Ch–6Fh | PORT2SC | Port 2 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 70h–73h | PORT3SC | Port 3 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 74h–77h | PORT4SC | Port 4 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 78h–7Bh | PORT5SC | Port 5 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 7Ch–7Fh | PORT6SC | Port 6 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 80h–83h | PORT7SC | Port 7 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 84h–9Fh | — | **Reserved** | Undefined | | RO |
| A0h–B3h | — | Debug Port Registers | Undefined | | See register description |
| B4h–3FFh | — | **Reserved** | Undefined | | RO |

***Note:*** Software must read and write these registers using only DWord accesses.These registers are divided into two sets. The first set at offsets MEM_BASE + 00:3Bh are implemented in the core power well. Unless otherwise noted, the core well registers are reset by the assertion of any of the following:

- Core well hardware reset
- HCRESET



- D3-to-D0 reset

The second set at offsets MEM_BASE + 60h to the end of the implemented register space are implemented in the Suspend power well. Unless otherwise noted, the suspend well registers are reset by the assertion of either of the following:

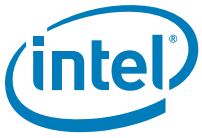
- Suspend well hardware reset
- HCRESET

### 16.2.2.1 USB2.0_CMD—USB 2.0 Command Register

Offset: MEM_BASE + 20–23h Attribute: R/W, RO
Default Value: 00080000h Size: 32 bits

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:24</td><td>Reserved. These bits are reserved and should be set to 0 when writing this register.</td></tr>
<tr><td>23:16</td><td>Interrupt Threshold Control — R/W. System software uses this field to select the maximum rate at which the host controller will issue interrupts. The only valid values are defined below. If software writes an invalid value to this register, the results are undefined.
<table>
<tr><th>Value</th><th>Maximum Interrupt Interval</th></tr>
<tr><td>00h</td><td>Reserved</td></tr>
<tr><td>01h</td><td>1 micro-frame</td></tr>
<tr><td>02h</td><td>2 micro-frames</td></tr>
<tr><td>04h</td><td>4 micro-frames</td></tr>
<tr><td>08h</td><td>8 micro-frames (default, equates to 1 ms)</td></tr>
<tr><td>10h</td><td>16 micro-frames (2 ms)</td></tr>
<tr><td>20h</td><td>32 micro-frames (4 ms)</td></tr>
<tr><td>40h</td><td>64 micro-frames (8 ms)</td></tr>
</table></td></tr>
<tr><td>15:8</td><td>Reserved. These bits are reserved and should be set to 0 when writing this register.</td></tr>
<tr><td>11:8</td><td>Unimplemented Asynchronous Park Mode Bits. Hardwired to 000b indicating the host controller does not support this optional feature.</td></tr>
<tr><td>7</td><td>Light Host Controller Reset — RO. Hardwired to 0. The Chipset does not implement this optional reset.</td></tr>
<tr><td>6</td><td>Interrupt on Async Advance Doorbell — R/W. This bit is used as a doorbell by software to tell the host controller to issue an interrupt the next time it advances asynchronous schedule.<br>0 = The host controller sets this bit to a 0 after it has set the Interrupt on Async Advance status bit (D29:F7:CAPLENGTH + 24h, bit 5) in the USB2.0_STS register to a 1.<br>1 = Software must write a 1 to this bit to ring the doorbell. When the host controller has evicted all appropriate cached schedule state, it sets the Interrupt on Async Advance status bit in the USB2.0_STS register. If the Interrupt on Async Advance Enable bit in the USB2.0_INTR register (D29:F7:CAPLENGTH + 28h, bit 5) is a 1 then the host controller will assert an interrupt at the next interrupt threshold. See the EHCI specification for operational details.<br><br>NOTE: Software should not write a 1 to this bit when the asynchronous schedule is inactive. Doing so will yield undefined results.</td></tr>
</table>

| Bit | Description |
|---|---|
| 5 | **Asynchronous Schedule Enable** — R/W. Default 0b. This bit controls whether the host controller skips processing the Asynchronous Schedule.<br>0 = Do not process the Asynchronous Schedule<br>1 = Use the ASYNCLISTADDR register to access the Asynchronous Schedule. |
| 4 | **Periodic Schedule Enable** — R/W. Default 0b. This bit controls whether the host controller skips processing the Periodic Schedule.<br>0 = Do not process the Periodic Schedule<br>1 = Use the PERIODICLISTBASE register to access the Periodic Schedule. |
| 3:2 | **Frame List Size** — RO. The Chipset hardwires this field to 00b because it only supports the<br>1024-element frame list size. |
| 1 | **Host Controller Reset (HCRESET)** — R/W. This control bit used by software to reset the host controller. The effects of this on root hub registers are similar to a Chip Hardware Reset<br>(i.e., RSMRST# assertion and PWROK deassertion on the Chipset).<br>When software writes a 1 to this bit, the host controller resets its internal pipelines, timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. A USB reset is not driven on downstream ports.<br><br>**NOTE:** PCI configuration registers and Host controller capability registers are not effected by this reset.<br>All operational registers, including port registers and port state machines are set to their initial values. Port ownership reverts to the companion host controller(s), with the side effects described in the EHCI spec. Software must re-initialize the host controller in order to return the host controller to an operational state.<br>This bit is set to 0 by the host controller when the reset process is complete. Software cannot terminate the reset process early by writing a 0 to this register.<br>Software should not set this bit to a 1 when the HCHalted bit (D29:F7:CAPLENGTH + 24h, bit 12) in the USB2.0_STS register is a 0. Attempting to reset an actively running host controller will result in undefined behavior. This reset me be used to leave EHCI port test modes. |



| Bit | Description |
|---|---|
| 0 | **Run/Stop (RS)** — R/W.<br>0 = Stop (default)<br>1 = Run. When set to a 1, the Host controller proceeds with execution of the schedule. The Host controller continues execution as long as this bit is set. When this bit is set to 0, the Host controller completes the current transaction on the USB and then halts. The HCHalted bit in the USB2.0_STS register indicates when the Host controller has finished the transaction and has entered the stopped state.<br>Software should not write a 1 to this field unless the host controller is in the Halted state<br>(i.e., HCHalted in the USBSTS register is a 1). The Halted bit is cleared immediately when the Run bit is set.<br>The following table explains how the different combinations of Run and Halted should be interpreted:<br><br>Memory read cycles initiated by the EHC that receive any status other than Successful will result in this bit being cleared. |

| Run/Stop | Halted | Interpretation |
|---|---|---|
| 0b | 0b | In the process of halting |
| 0b | 1b | Halted |
| 1b | 0b | Running |
| 1b | 1b | Invalid - the HCHalted bit clears immediately |

**NOTE:** The Command Register indicates the command to be executed by the serial bus host controller. Writing to the register causes a command to be executed.

### 16.2.2.2 USB2.0_STS—USB 2.0 Status Register

Offset: MEM_BASE + 24h–27h Attribute: R/WC, RO
Default Value: 00001000h Size: 32 bits

This register indicates pending interrupts and various states of the Host controller. The status resulting from a transaction on the serial bus is not indicated in this register. See the Interrupts description in section 4 of the EHCI specification for additional information concerning USB 2.0 interrupt conditions.

***Note:*** For the writable bits, software must write a 1 to clear bits that are set. Writing a 0 has no effect.

| Bit | Description |
|---|---|
| 31:16 | **Reserved**. These bits are reserved and should be set to 0 when writing this register. |
| 15 | **Asynchronous Schedule Status** — RO. This bit reports the current real status of the Asynchronous Schedule.<br>0 = Status of the Asynchronous Schedule is disabled. (Default)<br>1 = Status of the Asynchronous Schedule is enabled.<br><br>**NOTE:** The Host controller is not required to *immediately* disable or enable the Asynchronous Schedule when software transitions the *Asynchronous Schedule Enable* bit (D29:F7:CAPLENGTH + 20h, bit 5) in the USB2.0_CMD register. When this bit and the *Asynchronous Schedule Enable* bit are the same value, the Asynchronous Schedule is either enabled (1) or disabled (0). |

| Bit | Description |
|---|---|
| 14 | **Periodic Schedule Status** — RO. This bit reports the current real status of the Periodic Schedule.<br>0 = Status of the Periodic Schedule is disabled. (Default)<br>1 = Status of the Periodic Schedule is enabled.<br><br>**NOTE:** The Host controller is not required to *immediately* disable or enable the Periodic Schedule when software transitions the *Periodic Schedule Enable* bit (D29:F7:CAPLENGTH + 20h, bit 4) in the USB2.0_CMD register. When this bit and the *Periodic Schedule Enable* bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0). |
| 13 | **Reclamation** — RO. 0=Default. This read-only status bit is used to detect an empty asynchronous schedule. The operational model and valid transitions for this bit are described in Section 4 of the EHCI Specification. |
| 12 | **HCHalted** — RO.<br>0 = This bit is a 0 when the Run/Stop bit is a 1.<br>1 = The Host controller sets this bit to 1 after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host controller hardware (e.g., internal error). (Default) |
| 11:6 | **Reserved** |
| 5 | **Interrupt on Async Advance** — R/WC. 0=Default. System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a 1 to the *Interrupt on Async Advance Doorbell* bit (D29:F7:CAPLENGTH + 20h, bit 6) in the USB2.0_CMD register. This bit indicates the assertion of that interrupt source. |
| 4 | **Host System Error** — R/WC.<br>0 = No serious error occurred during a host system access involving the Host controller module<br>1 = The Host controller sets this bit to 1 when a serious error occurs during a host system access involving the Host controller module. A hardware interrupt is generated to the system. Memory read cycles initiated by the EHC that receive any status other than Successful will result in this bit being set.<br><br>When this error occurs, the Host controller clears the Run/Stop bit in the USB2.0_CMDregister (D29:F7:CAPLENGTH + 20h, bit 0) to prevent further execution of the scheduled TDs. A hardware interrupt is generated to the system (if enabled in the Interrupt Enable Register). |
| 3 | **Frame List Rollover** — R/WC.<br>0 = No *Frame List Index* rollover from its maximum value to 0.<br>1 = The Host controller sets this bit to a 1 when the *Frame List Index* (see Section) rolls over from its maximum value to 0. Since the Chipset only supports the 1024-entry Frame List Size, the *Frame List Index* rolls over every time FRNUM13 toggles. |

| Bit | Description |
|---|---|
| 14 | **Periodic Schedule Status** — RO. This bit reports the current real status of the Periodic Schedule.<br>0 = Status of the Periodic Schedule is disabled. (Default)<br>1 = Status of the Periodic Schedule is enabled.<br><br>**NOTE:** The Host controller is not required to *immediately* disable or enable the Periodic Schedule when software transitions the *Periodic Schedule Enable* bit (D29:F7:CAPLENGTH + 20h, bit 4) in the USB2.0_CMD register. When this bit and the *Periodic Schedule Enable* bit are the same value, the Periodic Schedule is either enabled (1) or disabled (0). |
| 13 | **Reclamation** — RO. 0=Default. This read-only status bit is used to detect an empty asynchronous schedule. The operational model and valid transitions for this bit are described in Section 4 of the EHCI Specification. |
| 12 | **HCHalted** — RO.<br>0 = This bit is a 0 when the Run/Stop bit is a 1.<br>1 = The Host controller sets this bit to 1 after it has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host controller hardware (e.g., internal error). (Default) |
| 11:6 | **Reserved** |
| 5 | **Interrupt on Async Advance** — R/WC. 0=Default. System software can force the host controller to issue an interrupt the next time the host controller advances the asynchronous schedule by writing a 1 to the *Interrupt on Async Advance Doorbell* bit (D29:F7:CAPLENGTH + 20h, bit 6) in the USB2.0_CMD register. This bit indicates the assertion of that interrupt source. |
| 4 | **Host System Error** — R/WC.<br>0 = No serious error occurred during a host system access involving the Host controller module<br>1 = The Host controller sets this bit to 1 when a serious error occurs during a host system access involving the Host controller module. A hardware interrupt is generated to the system. Memory read cycles initiated by the EHC that receive any status other than Successful will result in this bit being set.<br><br>When this error occurs, the Host controller clears the Run/Stop bit in the USB2.0_CMDregister (D29:F7:CAPLENGTH + 20h, bit 0) to prevent further execution of the scheduled TDs. A hardware interrupt is generated to the system (if enabled in the Interrupt Enable Register). |
| 3 | **Frame List Rollover** — R/WC.<br>0 = No *Frame List Index* rollover from its maximum value to 0.<br>1 = The Host controller sets this bit to a 1 when the *Frame List Index* (see Section) rolls over from its maximum value to 0. Since the Chipset only supports the 1024-entry Frame List Size, the *Frame List Index* rolls over every time FRNUM13 toggles. |

| Bit | Description |
|---|---|
| 2 | **Port Change Detect** — R/WC. This bit is allowed to be maintained in the Auxiliary power well. Alternatively, it is also acceptable that on a D3 to D0 transition of the EHCI HC device, this bit is loaded with the OR of all of the PORTSC change bits (including: Force port resume, overcurrent change, enable/disable change and connect status change). Regardless of the implementation, when this bit is readable (i.e., in the D0 state), it must provide a valid view of the Port Status registers.<br>0 = No change bit transition from a 0 to 1 or No Force Port Resume bit transition from 0 to 1 as a result of a J-K transition detected on a suspended port.<br>1 = The Host controller sets this bit to 1 when any port for which the *Port Owner* bit is set to 0 has a change bit transition from a 0 to 1 or a Force Port Resume bit transition from 0 to 1 as a result of a J-K transition detected on a suspended port. |
| 1 | **USB Error Interrupt (USBERRINT)** — R/WC.<br>0 = No error condition.<br>1 = The Host controller sets this bit to 1 when completion of a USB transaction results in an error condition (e.g., error counter underflow). If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and Bit 0 are set. See the EHCI specification for a list of the USB errors that will result in this interrupt being asserted. |
| 0 | **USB Interrupt (USBINT)** — R/WC.<br>0 = No completion of a USB transaction whose Transfer Descriptor had its IOC bit set. No short packet is detected.<br>1 = The Host controller sets this bit to 1 when the cause of an interrupt is a completion of a USB transaction whose Transfer Descriptor had its IOC bit set.<br>The Host controller also sets this bit to 1 when a short packet is detected (actual number of bytes received was less than the expected number of bytes). |

### 16.2.2.3 USB2.0_INTR—USB 2.0 Interrupt Enable Register

| | | | |
|---|---|---|---|
| Offset: | MEM_BASE + 28h–2Bh | Attribute: | R/W |
| Default Value: | 00000000h | Size: | 32 bits |

This register enables and disables reporting of the corresponding interrupt to the software. When a bit is set and the corresponding interrupt is active, an interrupt is generated to the host. Interrupt sources that are disabled in this register still appear in the USB2.0_STS Register to allow the software to poll for events. Each interrupt enable bit description indicates whether it is dependent on the interrupt threshold mechanism (see Section 4 of the EHCI specification), or not.

| Bit | Description |
|---|---|
| 31:6 | **Reserved**. These bits are reserved and should be 0 when writing this register. |
| 5 | **Interrupt on Async Advance Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1, and the Interrupt on Async Advance bit (D29:F7:CAPLENGTH + 24h, bit 5) in the USB2.0_STS register is a 1, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software clearing the Interrupt on Async Advance bit. |

| Bit | Description |
|---|---|
| 4 | **Host System Error Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1, and the Host System Error Status bit (D29:F7:CAPLENGTH + 24h, bit 4) in the USB2.0_STS register is a 1, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Host System Error bit. |
| 3 | **Frame List Rollover Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1, and the Frame List Rollover bit (D29:F7:CAPLENGTH + 24h, bit 3) in the USB2.0_STS register is a 1, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Frame List Rollover bit. |
| 2 | **Port Change Interrupt Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1, and the Port Change Detect bit (D29:F7:CAPLENGTH + 24h, bit 2) in the USB2.0_STS register is a 1, the host controller will issue an interrupt. The interrupt is acknowledged by software clearing the Port Change Detect bit. |
| 1 | **USB Error Interrupt Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1, and the USBERRINT bit (D29:F7:CAPLENGTH + 24h, bit 1) in the USB2.0_STS register is a 1, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software by clearing the USBERRINT bit in the USB2.0_STS register. |
| 0 | **USB Interrupt Enable** — R/W.<br>0 = Disable.<br>1 = Enable. When this bit is a 1, and the USBINT bit (D29:F7:CAPLENGTH + 24h, bit 0) in the USB2.0_STS register is a 1, the host controller will issue an interrupt at the next interrupt threshold. The interrupt is acknowledged by software by clearing the USBINT bit in the USB2.0_STS register. |

### 16.2.2.4 FRINDEX—Frame Index Register

| | | | |
|---|---|---|---|
| Offset: | MEM_BASE + 2Ch–2Fh | Attribute: | R/W |
| Default Value: | 00000000h | Size: | 32 bits |

The SOF frame number value for the bus SOF token is derived or alternatively managed from this register. Refer to Section 4 of the EHCI specification for a detailed explanation of the SOF value management requirements on the host controller. The value of FRINDEX must be within 125 µs
(1 micro-frame) ahead of the SOF token value. The SOF value may be implemented as an 11-bit shadow register. For this discussion, this shadow register is 11 bits and is named SOFV. SOFV updates every 8 micro-frames. (1 millisecond). An example implementation to achieve this behavior is to increment SOFV each time the FRINDEX[2:0] increments from 0 to 1.

Software must use the value of FRINDEX to derive the current micro-frame number, both for
high-speed isochronous scheduling purposes and to provide the **get** micro-frame number function required to client drivers. Therefore, the value of FRINDEX and the

value of SOFV must be kept consistent if chip is reset or software writes to FRINDEX. Writes to FRINDEX must also
**write-through** FRINDEX[13:3] to SOFV[10:0]. In order to keep the update as simple as possible, software should not write a FRINDEX value where the three least significant bits are 111b or 000b.

***Note:*** This register is used by the host controller to index into the periodic frame list. The register updates every 125 microseconds (once each micro-frame). Bits [12:3] are used to select a particular entry in the Periodic Frame List during periodic schedule execution. The number of bits used for the index is fixed at 10 for the Chipset since it only supports 1024-entry frame lists. This register must be written as a DWord. Word and byte writes produce undefined results. This register cannot be written unless the Host controller is in the Halted state as indicated by the *HCHalted* bit (D29:F7:CAPLENGTH + 24h, bit 12). A write to this register while the Run/Stop bit (D29:F7:CAPLENGTH + 20h, bit 0) is set to a 1 (USB2.0_CMD register) produces undefined results. Writes to this register also effect the SOF value. See Section 4 of the EHCI specification for details.

| Bit | Description |
|---|---|
| 31:14 | **Reserved** |
| 13:0 | **Frame List Current Index/Frame Number** — R/W. The value in this register increments at the end of each time frame (e.g., micro-frame).<br>Bits [12:3] are used for the Frame List current index. This means that each location of the frame list is accessed 8 times (frames or micro-frames) before moving to the next index. |

### 16.2.2.5 CTRLDSSEGMENT—Control Data Structure Segment Register

| | | | |
|---|---|---|---|
| Offset: | MEM_BASE + 30h–33h | Attribute: | R/W, RO |
| Default Value: | 00000000h | Size: | 32 bits |

This 32-bit register corresponds to the most significant address bits [63:32] for all EHCI data structures. Since the Chipset hardwires the 64-bit Addressing Capability field in HCCPARAMS to 1, then this register is used with the link pointers to construct 64-bit addresses to EHCI control data structures. This register is concatenated with the link pointer from either the PERIODICLISTBASE, ASYNCLISTADDR, or any control data structure link field to construct a 64-bit address. This register allows the host software to locate all control data structures within the same 4 GB memory segment.

| Bit | Description |
|---|---|
| 31:12 | **Upper Address[63:44]** — RO. Hardwired to 0s. The Chipset EHC is only capable of generating addresses up to 16 terabytes (44 bits of address). |
| 11:0 | **Upper Address[43:32]** — R/W. This 12-bit field corresponds to address bits 43:32 when forming a control data structure address. |

### 16.2.2.6 PERIODICLISTBASE—Periodic Frame List Base Address Register

| | | | |
|---|---|---|---|
| Offset: | MEM_BASE + 34h–37h | Attribute: | R/W |
| Default Value: | 00000000h | Size: | 32 bits |

This 32-bit register contains the beginning address of the Periodic Frame List in the system memory. Since the Chipset host controller operates in 64-bit mode (as indicated by the 1 in the 64-bit Addressing Capability field in the HCCSPARAMS register) (offset 08h, bit 0), then the most significant 32 bits of every control data structure address comes from the CTRLDSSEGMENT register. HCD loads this register prior to starting the schedule execution by the host controller. The memory structure referenced by this physical memory pointer is assumed to be 4-Kbyte aligned. The contents of this register are combined with the Frame Index Register (FRINDEX) to enable the Host controller to step through the Periodic Frame List in sequence.

| Bit | Description |
|---|---|
| 31:12 | **Base Address (Low)** — R/W. These bits correspond to memory address signals [31:12], respectively. |
| 11:0 | **Reserved**. Must be written as 0s. During runtime, the value of these bits are undefined. |

### 16.2.2.7 ASYNCLISTADDR—Current Asynchronous List Address Register

| | | | |
|---|---|---|---|
| Offset: | MEM_BASE + 38h–3Bh | Attribute: | R/W |
| Default Value: | 00000000h | Size: | 32 bits |

This 32-bit register contains the address of the next asynchronous queue head to be executed. Since the Chipset host controller operates in 64-bit mode (as indicated by a 1 in 64-bit Addressing Capability field in the HCCPARAMS register) (offset 08h, bit 0), then the most significant 32 bits of every control data structure address comes from the CTRLDSSEGMENT register (offset 08h). Bits [4:0] of this register cannot be modified by system software and will always return 0's when read. The memory structure referenced by this physical memory pointer is assumed to be 32-byte aligned.

| Bit | Description |
|---|---|
| 31:5 | **Link Pointer Low (LPL)** — R/W. These bits correspond to memory address signals [31:5], respectively. This field may only reference a Queue Head (QH). |
| 4:0 | **Reserved**. These bits are reserved and their value has no effect on operation. |

### 16.2.2.8 CONFIGFLAG—Configure Flag Register

| | | | |
|---|---|---|---|
| Offset: | MEM_BASE + 60h–63h | Attribute: | R/W |
| Default Value: | 00000000h | Size: | 32 bits |

This register is in the suspend power well. It is only reset by hardware when the suspend power is initially applied or in response to a host controller reset.

| Bit | Description |
|---|---|
| 31:1 | **Reserved**. Read from this field will always return 0. |
| 0 | **Configure Flag (CF)** — R/W. Host software sets this bit as the last action in its process of configuring the Host controller. This bit controls the default port-routing control logic. Bit values and side-effects are listed below. See section 4 of the EHCI spec for operation details.<br>0 = Port routing control logic default-routes each port to the classic host controllers (default).<br>1 = Port routing control logic default-routes all ports to this host controller. |

### 16.2.2.9 PORTSC—Port N Status and Control Register

| | | | |
|---|---|---|---|
| Offset: | Port 0: MEM_BASE + 64h–67h<br>Port 1: MEM_BASE + 68–6Bh<br>Port 2: MEM_BASE + 6C–6Fh<br>Port 3: MEM_BASE + 70–73h<br>Port 4: MEM_BASE + 74–77h<br>Port 5: MEM_BASE + 78–7Bh<br>Port 6: MEM_BASE + 7C–7Fh<br>Port 7: MEM_BASE + 80–83h | | |
| Attribute: | R/W, R/WC, RO | | |
| Default Value: | 00003000h | Size: | 32 bits |

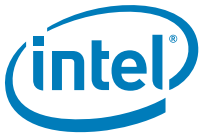
A host controller must implement one or more port registers. Software uses the N_Port information from the Structural Parameters Register to determine how many ports need to be serviced. All ports have the structure defined below. Software must not write to unreported Port Status and Control Registers.

This register is in the suspend power well. It is only reset by hardware when the suspend power is initially applied or in response to a host controller reset. The initial conditions of a port are:

- No device connected
- Port disabled.

When a device is attached, the port state transitions to the attached state and system software will process this as with any status change notification. Refer to Section 4 of the EHCI specification for operational requirements for how change events interact with port suspend mode.

| Bit | Description |
|---|---|
| 31:23 | **Reserved**. These bits are reserved for future use and will return a value of 0's when read. |



<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>22</td><td>Wake on Overcurrent Enable (WKOC_E) — R/W.<br>0 = Disable. (Default)<br>1 = Enable. Writing this bit to a 1 enables the setting of the PME Status bit in the Power Management Control/Status Register (offset 54, bit 15) when the overcurrent Active bit (bit 4 of this register) is set.</td></tr>
<tr><td>21</td><td>Wake on Disconnect Enable (WKDSCNNT_E) — R/W.<br>0 = Disable. (Default)<br>1 = Enable. Writing this bit to a 1 enables the setting of the PME Status bit in the Power Management Control/Status Register (offset 54, bit 15) when the Current Connect Status changes from connected to disconnected (i.e., bit 0 of this register changes from 1 to 0).</td></tr>
<tr><td>20</td><td>Wake on Connect Enable (WKCNNT_E) — R/W.<br>0 = Disable. (Default)<br>1 = Enable. Writing this bit to a 1 enables the setting of the PME Status bit in the Power Management Control/Status Register (offset 54, bit 15) when the Current Connect Status changes from disconnected to connected (i.e., bit 0 of this register changes from 0 to 1).</td></tr>
<tr><td>19:16</td><td>Port Test Control — R/W. When this field is 0's, the port is NOT operating in a test mode. A non-zero value indicates that it is operating in test mode and the specific test mode is indicated by the specific value. The encoding of the test mode bits are (0110b – 1111b are reserved):<br><br>
<table>
<tr><th>Value</th><th>Maximum Interrupt Interval</th></tr>
<tr><td>0000b</td><td>Test mode not enabled (default)</td></tr>
<tr><td>0001b</td><td>Test J_STATE</td></tr>
<tr><td>0010b</td><td>Test K_STATE</td></tr>
<tr><td>0011b</td><td>Test SE0_NAK</td></tr>
<tr><td>0100b</td><td>Test Packet</td></tr>
<tr><td>0101b</td><td>FORCE_ENABLE</td></tr>
</table>
Refer to USB Specification Revision 2.0, Chapter 7 for details on each test mode.</td></tr>
<tr><td>15:14</td><td>Reserved — R/W. Should be written to =00b.</td></tr>
<tr><td>13</td><td>Port Owner — R/W. Default = 1b. This bit unconditionally goes to a 0 when the Configured Flag bit in the USB2.0_CMD register makes a 0 to 1 transition.<br>System software uses this field to release ownership of the port to a selected host controller (in the event that the attached device is not a high-speed device). Software writes a 1 to this bit when the attached device is not a high-speed device. A 1 in this bit means that a companion host controller owns and controls the port. See Section 4 of the EHCI Specification for operational details.</td></tr>
<tr><td>12</td><td>Port Power (PP) — RO. Read-only with a value of 1. This indicates that the port does have power.</td></tr>
<tr><td>11:10</td><td>Line Status— RO.These bits reflect the current logical levels of the D+ (bit 11) and D– (bit 10) signal lines. These bits are used for detection of low-speed USB devices prior to the port reset and enable sequence. This field is valid only when the port enable bit is 0 and the current connect status bit is set to a 1.<br>00 = SE0<br>10 = J-state<br>01 = K-state<br>11 = Undefined</td></tr>
<tr><td>9</td><td>Reserved. This bit will return a 0 when read.</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>8</td><td>Port Reset — R/W. Default = 0. When software writes a 1 to this bit (from a 0), the bus reset sequence as defined in the USB Specification, Revision 2.0 is started. Software writes a 0 to this bit to terminate the bus reset sequence. Software must keep this bit at a 1 long enough to ensure the reset sequence completes as specified in the USB Specification, Revision 2.0.<br>1 = Port is in Reset.<br>0 = Port is not in Reset.<br>NOTE: When software writes a 0 to this bit, there may be a delay before the bit status changes to a 0. The bit status will not read as a 0 until after the reset has completed. If the port is in high-speed mode after reset is complete, the host controller will automatically enable this port (e.g., set the Port Enable bit to a 1). A host controller must terminate the reset and stabilize the state of the port within 2 milliseconds of software transitioning this bit from 0 to 1.<br><br>For example: if the port detects that the attached device is high-speed during reset, then the host controller must have the port in the enabled state within 2 ms of software writing this bit to a 0. The HCHalted bit (D29:F7:CAPLENGTH + 24h, bit 12) in the USB2.0_STS register should be a 0 before software attempts to use this bit. The host controller may hold Port Reset asserted to a 1 when the HCHalted bit is a 1. This bit is 0 if Port Power is 0.<br>NOTE: System software should not attempt to reset a port if the HCHalted bit in the USB2.0_STS register is a 1. Doing so will result in undefined behavior.</td></tr>
<tr><td>7</td><td>Suspend — R/W.<br>0 = Port not in suspend state.(Default)<br>1 = Port in suspend state.<br>Port Enabled Bit and Suspend bit of this register define the port states as follows:<br><br>
<table>
<tr><th>Port Enabled</th><th>Suspend</th><th>Port State</th></tr>
<tr><td>0</td><td>X</td><td>Disabled</td></tr>
<tr><td>1</td><td>0</td><td>Enabled</td></tr>
<tr><td>1</td><td>1</td><td>Suspend</td></tr>
</table>
When in suspend state, downstream propagation of data is blocked on this port, except for port reset. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port depending on the activity on the port.<br>The host controller will unconditionally set this bit to a 0 when software sets the Force Port Resume bit to a 0 (from a 1). A write of 0 to this bit is ignored by the host controller.<br>If host software sets this bit to a 1 when the port is not enabled (i.e., Port enabled bit is a 0) the results are undefined.</td></tr>
</table>

| Bit | Description |
|---|---|
| 6 | **Force Port Resume** — R/W.<br>0 = No resume (K-state) detected/driven on port. (Default)<br>1 = Resume detected/driven on port. Software sets this bit to a 1 to drive resume signaling. The Host controller sets this bit to a 1 if a J-to-K transition is detected while the port is in the Suspend state. When this bit transitions to a 1 because a J-to-K transition is detected, the Port Change Detect bit (D29:F7:CAPLENGTH + 24h, bit 2) in the USB2.0_STS register is also set to a 1. If software sets this bit to a 1, the host controller must not set the Port Change Detect bit.<br><br>**NOTE:** When the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the USB Specification, Revision 2.0. The resume signaling (Full-speed 'K') is driven on the port as long as this bit remains a 1. Software must appropriately time the Resume and set this bit to a 0 when the appropriate amount of time has elapsed. Writing a 0 (from 1) causes the port to return to high-speed mode (forcing the bus below the port into a high-speed idle). This bit will remain a 1 until the port has switched to the high-speed idle. |
| 5 | **Overcurrent Change** — R/WC. The functionality of this bit is not dependent upon the port owner. Software clears this bit by writing a 1 to it.<br>0 = No change. (Default)<br>1 = There is a change to Overcurrent Active. |
| 4 | **Overcurrent Active** — RO.<br>0 = This port does not have an overcurrent condition. (Default)<br>1 = This port currently has an overcurrent condition. This bit will automatically transition from 1 to 0 when the over current condition is removed. The Chipset automatically disables the port when the overcurrent active bit is 1. |
| 3 | **Port Enable / Disable Change** — R/WC. For the root hub, this bit gets set to a 1 only when a port is disabled due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification for the definition of a port error). This bit is not set due to the Disabled-to-Enabled transition, nor due to a disconnect. Software clears this bit by writing a 1 to it.<br>0 = No change in status. (Default).<br>1 = Port enabled/disabled status has changed. |
| 2 | **Port Enabled / Disabled** — R/W. Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a 1 to this bit. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events.<br>0 = Disable<br>1 = Enable (Default) |
| 1 | **Connect Status Change** — R/WC. This bit indicates a change has occurred in the port's Current Connect Status. Software sets this bit to 0 by writing a 1 to it.<br>0 = No change (Default).<br>1 = Change in Current Connect Status. The host controller sets this bit for all changes to the port device connect status, even if system software has not cleared an existing connect status change. For example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be "setting" an already-set bit (i.e., the bit will remain set). |

| Bit | Description |
|---|---|
| 0 | **Current Connect Status** — RO. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set.<br>0 = No device is present. (Default)<br>1 = Device is present on port. |

## 16.2.3 USB 2.0-Based Debug Port Register

The Debug port's registers are located in the same memory area, defined by the Base Address Register (MEM_BASE), as the standard EHCI registers. The base offset for the debug port registers (A0h) is declared in the Debug Port Base Offset Capability Register at Configuration offset 5Ah (D29:F7:offset 5Ah). The specific EHCI port that supports this debug capability (port 0) is indicated by a 4-bit field (bits 20–23) in the HCSPARAMS register of the EHCI controller. The address map of the Debug Port registers is shown in Table 16-143.

**Table 16-143.Debug Port Register Address Map**

| MEM_BASE + Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| A0–A3h | CNTL_STS | Control/Status | 00000000h | R/W, R/WC, RO, WO |
| A4–A7h | USBPID | USB PIDs | 00000000h | R/W, RO |
| A8–ABh | DATABUF[3:0] | Data Buffer (Bytes 3:0) | 00000000h | R/W |
| AC–AFh | DATABUF[7:4] | Data Buffer (Bytes 7:4) | 00000000h | R/W |
| B0–B3h | CONFIG | Configuration | 00007F01h | R/W |

**NOTES:**
1. All of these registers are implemented in the core well and reset by PLTRST#, EHC HCRESET, and a EHC D3-to-D0 transition.
2. The hardware associated with this register provides no checks to ensure that software programs the interface correctly. How the hardware behaves when programmed inappropriately is undefined.

### 16.2.3.1 CNTL_STS—Control/Status Register

Offset: MEM_BASE + A0h Attribute: R/W, R/WC, RO, WO
Default Value: 0000h Size: 32 bits

| Bit | Description |
|---|---|
| 31 | **Reserved** |
| 30 | **OWNER_CNT —** R/W.<br>0 = Ownership of the debug port is NOT forced to the EHCI controller (Default)<br>1 = Ownership of the debug port is forced to the EHCI controller (i.e. immediately taken away from the companion Classic USB Host controller) If the port was already owned by the EHCI controller, then setting this bit has no effect. This bit overrides all of the ownership-related bits in the standard EHCI registers. |

| Bit | Description |
|---|---|
| 29 | **Reserved** |
| 28 | **ENABLED_CNT —** R/W.<br>0 = Software can clear this by writing a 0 to it. The hardware clears this bit for the same conditions where the Port Enable/Disable Change bit (in the PORTSC register) is set. (Default)<br>1 = Debug port is enabled for operation. Software can directly set this bit if the port is already enabled in the associated PORTSC register (this is enforced by the hardware). |
| 27:17 | **Reserved** |
| 16 | **DONE_STS —** R/WC. Software can clear this by writing a 1 to it.<br>0 = Request Not complete<br>1 = Set by hardware to indicate that the request is complete. |
| 15:12 | LINK_ID_STS — RO. This field identifies the link interface.<br>0h = Hardwired. Indicates that it is a USB Debug Port. |
| 11 | **Reserved**. This bit returns 0 when read. Writes have no effect. |
| 10 | **IN_USE_CNT —** R/W. Set by software to indicate that the port is in use. Cleared by software to indicate that the port is free and may be used by other software. This bit is cleared after reset. (This bit has no effect on hardware.) |
| 9:7 | **EXCEPTION_STS —** RO. This field indicates the exception when the ERROR_GOOD#_STS bit is set. This field should be ignored if the ERROR_GOOD#_STS bit is 0.<br>000 =No Error. (Default)<br>Note: this should not be seen, since this field should only be checked if there is an error.<br>001 =Transaction error: indicates the USB 2.0 transaction had an error (CRC, bad PID, timeout, etc.)<br>010 =Hardware error. Request was attempted (or in progress) when port was suspended or reset.<br>All Other combinations are reserved |
| 6 | **ERROR_GOOD#_STS —** RO.<br>0 = Hardware clears this bit to 0 after the proper completion of a read or write. (Default)<br>1 = Error has occurred. Details on the nature of the error are provided in the Exception field. |
| 5 | **GO_CNT** — WO.<br>0 = Hardware clears this bit when hardware sets the DONE_STS bit. (Default)<br>1 = Causes hardware to perform a read or write request.<br><br>**NOTE:** Writing a 1 to this bit when it is already set may result in undefined behavior. |
| 4 | **WRITE_READ#_CNT —** R/W. Software clears this bit to indicate that the current request is a read. Software sets this bit to indicate that the current request is a write.<br>0 = Read (Default)<br>1 = Write |

| Bit | Description |
|---|---|
| 3:0 | **DATA_LEN_CNT —** R/W. This field is used to indicate the size of the data to be transferred.<br>default = 0h.<br>For write operations, this field is set by software to indicate to the hardware how many bytes of data in Data Buffer are to be transferred to the console. A value of 0h indicates that a zero-length packet should be sent. A value of 1–8 indicates 1–8 bytes are to be transferred. Values 9–Fh are invalid and how hardware behaves if used is undefined.<br>For read operations, this field is set by hardware to indicate to software how many bytes in Data Buffer are valid in response to a read operation. A value of 0h indicates that a zero length packet was returned and the state of Data Buffer is not defined. A value of 1–8 indicates 1–8 bytes were received. Hardware is not allowed to return values 9–Fh.<br>The transferring of data always starts with byte 0 in the data area and moves toward byte 7 until the transfer size is reached. |

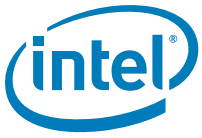
**NOTES:**
1. Software should do Read-Modify-Write operations to this register to preserve the contents of bits not being modified. This include **Reserved** bits.
2. To preserve the usage of RESERVED bits in the future, software should always write the same value read from the bit until it is defined. **Reserved** bits will always return 0 when read.

### 16.2.3.2 USBPID—USB PIDs Register

| | | | |
|---|---|---|---|
| Offset: | MEM_BASE + A4h | Attribute: | R/W, RO |
| Default Value: | 0000h | Size: | 32 bits |

This DWord register is used to communicate PID information between the USB debug driver and the USB debug port. The debug port uses some of these fields to generate USB packets, and uses other fields to return PID information to the USB debug driver.

| Bit | Description |
|---|---|
| 31:24 | **Reserved**: These bits will return 0 when read. Writes will have no effect. |
| 23:16 | **RECEIVED_PID_STS[23:16]** — RO. Hardware updates this field with the received PID for transactions in either direction. When the controller is writing data, this field is updated with the handshake PID that is received from the device. When the host controller is reading data, this field is updated with the data packet PID (if the device sent data), or the handshake PID (if the device NAKs the request). This field is valid when the hardware clears the GO_DONE#_CNT bit. |
| 15:8 | **SEND_PID_CNT[15:8]** — R/W. Hardware sends this PID to begin the data packet when sending data to USB (i.e., WRITE_READ#_CNT is asserted). Software typically sets this field to either DATA0 or DATA1 PID values. |
| 7:0 | **TOKEN_PID_CNT[7:0]** — R/W. Hardware sends this PID as the Token PID for each USB transaction. Software typically sets this field to either IN, OUT, or SETUP PID values. |



### 16.2.3.3 DATABUF[7:0]—Data Buffer Bytes[7:0] Register

| | | | |
|---|---|---|---|
| Offset: | MEM_BASE + A8h–AFh | Attribute: | R/W |
| Default Value: | 0000000000000000h | Size: | 64 bits |

This register can be accessed as 8 separate 8-bit registers or 2 separate 32-bit register.

| Bit | Description |
|---|---|
| 63:0 | **DATABUFFER[63:0]** — R/W. This field is the 8 bytes of the data buffer. Bits 7:0 correspond to least significant byte (byte 0). Bits 63:56 correspond to the most significant byte (byte 7).<br>The bytes in the Data Buffer must be written with data before software initiates a write request. For a read request, the Data Buffer contains valid data when DONE_STS bit (offset A0, bit 16) is cleared by the hardware, ERROR_GOOD#_STS (offset A0, bit 6) is cleared by the hardware, and the DATA_LENGTH_CNT field (offset A0, bits 3:0) indicates the number of bytes that are valid. |

### 16.2.3.4 CONFIG—Configuration Register

| | | | |
|---|---|---|---|
| Offset: | MEM_BASE + B0–B3h | Attribute: | R/W |
| Default Value: | 00007F01h | Size: | 32 bits |

| Bit | Description |
|---|---|
| 31:15 | **Reserved** |
| 14:8 | **USB_ADDRESS_CNF** — R/W. This 7-bit field identifies the USB device address used by the controller for all Token PID generation. (Default = 7Fh) |
| 7:4 | **Reserved** |
| 3:0 | **USB_ENDPOINT_CNF** — R/W. This 4-bit field identifies the endpoint used by the controller for all Token PID generation. (Default = 01h) |

§

# 17 SMBus Controller Registers (D31:F3)

## 17.1 PCI Configuration Registers (SMBUS—D31:F3)

**Table 17-144.SMBus Controller PCI Register Address Map (SMBUS—D31:F3)**

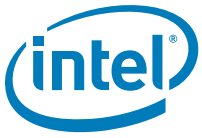
| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00h–01h | VID | Vendor Identification | 8086 | RO |
| 02h–03h | DID | Device Identification | See register description. | RO |
| 04h–05h | PCICMD | PCI Command | 0000h | R/W, RO |
| 06h–07h | PCISTS | PCI Status | 0280h | RO, R/WC |
| 08h | RID | Revision Identification | See register description. | RO |
| 09h | PI | Programming Interface | 00h | RO |
| 0Ah | SCC | Sub Class Code | 05h | RO |
| 0Bh | BCC | Base Class Code | 0Ch | RO |
| 20h–23h | SMB_BASE | SMBus Base Address | 00000001h | R/W, RO |
| 2Ch–2Dh | SVID | Subsystem Vendor Identification | 00h | RO |
| 2Eh–2Fh | SID | Subsystem Identification | 00h | R/WO |
| 3Ch | INT_LN | Interrupt Line | 00h | R/W |
| 3Dh | INT_PN | Interrupt Pin | See description | RO |
| 40h | HOSTC | Host Configuration | 00h | R/W |

**NOTE:** Registers that are not shown should be treated as **Reserved** (See Section 9.2 for details).

### 17.1.1 VID—Vendor Identification Register (SMBUS—D31:F3)

Address: 00h–01h Attribute: RO
Default Value: 8086h Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Vendor ID** — RO. This is a 16-bit value assigned to Intel |



### 17.1.2 DID—Device Identification Register (SMBUS—D31:F3)

Address: 02h–03h Attribute: RO
Default Value: See bit description Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Device ID** — RO. This is a 16-bit value assigned to the Chipset SMBus controller. |

### 17.1.3 PCICMD—PCI Command Register (SMBUS—D31:F3)

Address: 04h–05h Attributes: RO, R/W
Default Value: 0000h Size: 16 bits

| Bit | Description |
|---|---|
| 15:11 | **Reserved** |
| 10 | **Interrupt Disable** — R/W.<br>0 = Enable<br>1 = Disables SMBus to assert its PIRQB# signal. |
| 9 | Fast Back to Back Enable (FBE) — RO. Hardwired to 0. |
| 8 | **SERR# Enable (SERR_EN)** — R/W.<br>0 = Enables SERR# generation.<br>1 = Disables SERR# generation. |
| 7 | Wait Cycle Control (WCC) — RO. Hardwired to 0. |
| 6 | **Parity Error Response (PER)** — R/W.<br>0 = Disable<br>1 = Sets Detected Parity Error bit (D31:F3:06, bit 15) when a parity error is detected. |
| 5 | **VGA Palette Snoop (VPS)** — RO. Hardwired to 0. |
| 4 | **Postable Memory Write Enable (PMWE)** — RO. Hardwired to 0. |
| 3 | **Special Cycle Enable (SCE)** — RO. Hardwired to 0. |
| 2 | **Bus Master Enable (BME)** — RO. Hardwired to 0. |
| 1 | **Memory Space Enable (MSE)** — RO. Hardwired to 0. |
| 0 | **I/O Space Enable (IOSE)** — R/W.<br>0 = Disable<br>1 = Enables access to the SM Bus I/O space registers as defined by the Base Address Register. |

### 17.1.4 PCISTS—PCI Status Register (SMBUS—D31:F3)

Address: 06h–07h
Default Value: 0280h
Attributes: RO, R/WC
Size: 16 bits

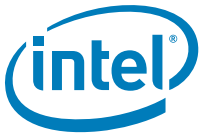
***Note:*** For the writable bits, software must write a 1 to clear bits that are set. Writing a 0 to the bit has no effect.

| Bit | Description |
|---|---|
| 15 | **Detected Parity Error (DPE)** — R/WC.<br>0 = No parity error detected.<br>1 = Parity error detected. |
| 14 | **Signaled System Error (SSE)** — R/WC.<br>0 = No system error detected.<br>1 = System error detected. |
| 13 | Received Master Abort (RMA) — RO. Hardwired to 0. |
| 12 | Received Target Abort (RTA) — RO. Hardwired to 0. |
| 11 | **Signaled Target Abort (STA)** — R/WC.<br>0 = Chipset did Not terminate transaction for this function with a target abort.<br>1 = The function is targeted with a transaction that the Chipset terminates with a target abort. |
| 10:9 | **DEVSEL# Timing Status (DEVT)** — RO. This 2-bit field defines the timing for DEVSEL# assertion for positive decode.<br>01 = Medium timing. |
| 8 | **Data Parity Error Detected (DPED)** — RO. Hardwired to 0. |
| 7 | **Fast Back to Back Capable (FB2BC)** — RO. Hardwired to 1. |
| 6 | **User Definable Features (UDF)** — RO. Hardwired to 0. |
| 5 | **66 MHz Capable (66MHZ_CAP)** — RO. Hardwired to 0. |
| 4 | **Capabilities List (CAP_LIST)** — RO. Hardwired to 0 because there are no capability list structures in this function |
| 3 | **Interrupt Status (INTS)** — RO. This bit indicates that an interrupt is pending. It is independent from the state of the Interrupt Enable bit in the PCI Command register. |
| 2:0 | **Reserved** |

### 17.1.5 RID—Revision Identification Register (SMBUS—D31:F3)

Offset Address: 08h
Default Value: See bit description
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Revision ID — RO. |



### 17.1.6 PI—Programming Interface Register (SMBUS—D31:F3)

Offset Address: 09h
Default Value: 00h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Reserved** |

### 17.1.7 SCC—Sub Class Code Register (SMBUS—D31:F3)

Address Offset: 0Ah
Default Value: 05h
Attributes: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Sub Class Code (SCC)** — RO.<br>05h = SM Bus serial controller |

### 17.1.8 BCC—Base Class Code Register (SMBUS—D31:F3)

Address Offset: 0Bh
Default Value: 0Ch
Attributes: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Base Class Code (BCC)** — RO.<br>0Ch = Serial controller. |

### 17.1.9 SMB_BASE—SMBUS Base Address Register (SMBUS—D31:F3)

Address Offset: 20–23h
Default Value: 00000001h
Attribute: R/W, RO
Size: 32-bits

| Bit | Description |
|---|---|
| 31:16 | **Reserved** — RO |
| 15:5 | **Base Address** — R/W. This field provides the 32-byte system I/O base address for the Chipset SMB logic. |
| 4:1 | **Reserved** — RO |
| 0 | IO Space Indicator — RO. Hardwired to 1 indicating that the SMB logic is I/O mapped. |

### 17.1.10 SVID — Subsystem Vendor Identification Register (SMBUS—D31:F2/F4)

| | | | |
|---|---|---|---|
| Address Offset: | 2Ch–2Dh | Attribute: | RO |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:0 | **Subsystem Vendor ID (SVID)** — RO. The SVID register, in combination with the Subsystem ID (SID) register, enables the operating system (OS) to distinguish subsystems from each other. The value returned by reads to this register is the same as that which was written by BIOS into the IDE SVID register.<br><br>**NOTE:** Software can write to this register only once per core well reset. Writes should be done as a single 16-bit cycle. |

### 17.1.11 SID — Subsystem Identification Register (SMBUS—D31:F2/F4)

| | | | |
|---|---|---|---|
| Address Offset: | 2Eh–2Fh | Attribute: | R/WO |
| Default Value: | 00h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:0 | **Subsystem ID (SID)** — RO. The SID register, in combination with the SVID register, enables the operating system (OS) to distinguish subsystems from each other. The value returned by reads to this register is the same as that which was written by BIOS into the IDE SID register.<br><br>**NOTE:** Software can write to this register only once per core well reset. Writes should be done as a single 16-bit cycle. |

### 17.1.12 INT_LN—Interrupt Line Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Address Offset: | 3Ch | Attributes: | R/W |
| Default Value: | 00h | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:0 | **Interrupt Line (INT_LN)** — R/W. This data is not used by the Chipset. It is to communicate to software the interrupt line that the interrupt pin is connected to PIRQB#. |

### 17.1.13 INT_PN—Interrupt Pin Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Address Offset: | 3Dh | Attributes: | RO |
| Default Value: | See description | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:0 | **Interrupt PIN (INT_PN)** — RO. This reflects the value of D31IP.SMIP in chipset configuration space. |

### 17.1.14 HOSTC—Host Configuration Register (SMBUS—D31:F3)

Address Offset: 40h
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bit | Description |
|---|---|
| 7:3 | **Reserved** |
| 2 | **I²C_EN** — R/W.<br>0 = SMBus behavior.<br>1 = The Chipset is enabled to communicate with I²C devices. This will change the formatting of some commands. |
| 1 | **SMB_SMI_EN** — R/W.<br>0 = SMBus interrupts will not generate an SMI#.<br>1 = Any source of an SMB interrupt will instead be routed to generate an SMI#. Refer to Section 5.21.4 - Volume 1 (Interrupts / SMI#). This bit needs to be set for SMBALERT# to be enabled. |
| 0 | **SMBus Host Enable (HST_EN)** — R/W.<br>0 = Disable the SMBus Host controller.<br>1 = Enable. The SMB Host controller interface is enabled to execute commands. The INTREN bit (offset SMBASE + 02h, bit 0) needs to be enabled for the SMB Host controller to interrupt or SMI#. Note that the SMB Host controller will not respond to any new requests until all interrupt requests have been cleared. |

## 17.2 SMBus I/O Registers

**Table 17-145.SMBus I/O Register Address Map**

| SMB_BASE + Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00h | HST_STS | Host Status | 00h | R/WC, RO, R/WC (special) |
| 02h | HST_CNT | Host Control | 00h | R/W, WO |
| 03h | HST_CMD | Host Command | 00h | R/W |
| 04h | XMIT_SLVA | Transmit Slave Address | 00h | R/W |
| 05h | HST_D0 | Host Data 0 | 00h | R/W |
| 06h | HST_D1 | Host Data 1 | 00h | R/W |
| 07h | HOST_BLOCK_DB | Host Block Data Byte | 00h | R/W |
| 08h | PEC | Packet Error Check | 00h | R/W |
| 09h | RCV_SLVA | Receive Slave Address | 44h | R/W |
| 0Ah–0Bh | SLV_DATA | Receive Slave Data | 0000h | RO |
| 0Ch | AUX_STS | Auxiliary Status | 00h | R/WC, RO |
| 0Dh | AUX_CTL | Auxiliary Control | 00h | R/W |
| 0Eh | SMLINK_PIN_CTL | SMLink Pin Control (TCO Compatible Mode) | See register description | R/W, RO |

**Table 17-145.SMBus I/O Register Address Map**

| SMB_BASE + Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 0Fh | SMBUS_PIN_CTL | SMBus Pin Control | See register description | R/W, RO |
| 10h | SLV_STS | Slave Status | 00h | R/WC |
| 11h | SLV_CMD | Slave Command | 00h | R/W |
| 14h | NOTIFY_DADDR | Notify Device Address | 00h | RO |
| 16h | NOTIFY_DLOW | Notify Data Low Byte | 00h | RO |
| 17h | NOTIFY_DHIGH | Notify Data High Byte | 00h | RO |

## 17.2.1 HST_STS—Host Status Register (SMBUS—D31:F3)

Register Offset: SMBASE + 00h
Attribute: R/WC, R/WC (special), RO
Default Value: 00h
Size: 8-bits

All status bits are set by hardware and cleared by the software writing a one to the particular bit position. Writing a 0 to any bit position has no effect.

| Bit | Description |
|---|---|
| 7 | **Byte Done Status (DS)** — R/WC.<br>0 = Software can clear this by writing a 1 to it.<br>1 = Host controller received a byte (for Block Read commands) or if it has completed transmission of a byte (for Block Write commands) when the 32-byte buffer is not being used. Note that this bit will be set, even on the last byte of the transfer. This bit is not set when transmission is due to the LAN interface heartbeat.<br>This bit has no meaning for block transfers when the 32-byte buffer is enabled.<br><br>**NOTE:** When the last byte of a block message is received, the host controller will set this bit. However, it will not immediately set the INTR bit (bit 1 in this register). When the interrupt handler clears the DS bit, the message is considered complete, and the host controller will then set the INTR bit (and generate another interrupt). Thus, for a block message of n bytes, the Chipset will generate n+1 interrupts. The interrupt handler needs to be implemented to handle these cases. |
| 6 | **INUSE_STS** — R/WC (special). This bit is used as semaphore among various independent software threads that may need to use the chipset's SMBus logic, and has no other effect on hardware.<br>0 = After a full PCI reset, a read to this bit returns a 0.<br>1 = After the first read, subsequent reads will return a 1. A write of a 1 to this bit will reset the next read value to 0. Writing a 0 to this bit has no effect. Software can poll this bit until it reads a 0, and will then own the usage of the host controller. |
| 5 | **SMBALERT_STS** — R/WC.<br>0 = Interrupt or SMI# was not generated by SMBALERT#. Software clears this bit by writing a 1 to it.<br>1 = The source of the interrupt or SMI# was the SMBALERT# signal. This bit is only cleared by software writing a 1 to the bit position or by RSMRST# going low.<br>If the signal is programmed as a GPIO, then this bit will not be set. |

| Bit | Description |
|---|---|
| 4 | **FAILED** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = The source of the interrupt or SMI# was a failed bus transaction. This bit is set in response to the KILL bit being set to terminate the host transaction. |
| 3 | **BUS_ERR** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = The source of the interrupt of SMI# was a transaction collision. |
| 2 | **DEV_ERR** — R/WC.<br>0 = Software clears this bit by writing a 1 to it. The Chipset will then deassert the interrupt or SMI#.<br>1 = The source of the interrupt or SMI# was due to one of the following:<br>• Invalid Command Field,<br>• Unclaimed Cycle (host initiated),<br>• Host Device Time-out Error. |
| 1 | **INTR** — R/WC (special). This bit can only be set by termination of a command. INTR is not dependent on the INTREN bit (offset SMBASE + 02h, bit 0) of the Host controller register (offset 02h). It is only dependent on the termination of the command. If the INTREN bit is not set, then the INTR bit will be set, although the interrupt will not be generated. Software can poll the INTR bit in this non-interrupt case.<br>0 = Software clears this bit by writing a 1 to it. The Chipset then deasserts the interrupt or SMI#.<br>1 = The source of the interrupt or SMI# was the successful completion of its last command. |
| 0 | **HOST_BUSY** — RO.<br>0 = Cleared by the Chipset when the current transaction is completed.<br>1 = Indicates that the Chipset is running a command from the host interface. No SMB registers should be accessed while this bit is set, except the BLOCK DATA BYTE Register. The BLOCK DATA BYTE Register can be accessed when this bit is set only when the SMB_CMD bits in the Host Control Register are programmed for Block command or $I^2C$ Read command. This is necessary in order to check the DONE_STS bit. |

### 17.2.2 HST_CNT—Host Control Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | SMBASE + 02h | Attribute: | R/W, WO |
| Default Value: | 00h | Size: | 8-bits |

***Note:*** A read to this register will clear the byte pointer of the 32-byte buffer.

| Bit | Description |
|---|---|
| 7 | **PEC_EN** — R/W.<br>0 = SMBus host controller does not perform the transaction with the PEC phase appended.<br>1 = Causes the host controller to perform the SMBus transaction with the Packet Error Checking phase appended. For writes, the value of the PEC byte is transferred from the PEC Register. For reads, the PEC byte is loaded in to the PEC Register. This bit must be written prior to the write in which the **START** bit is set. |
| 6 | **START** — WO.<br>0 = This bit will always return 0 on reads. The HOST_BUSY bit in the Host Status register (offset 00h) can be used to identify when the Chipset has finished the command.<br>1 = Writing a 1 to this bit initiates the command described in the SMB_CMD field. All registers should be setup prior to writing a 1 to this bit position. |
| 5 | **LAST_BYTE** — WO. This bit is used for Block Read commands.<br><br>1 = Software sets this bit to indicate that the next byte will be the last byte to be received for the block. This causes the Chipset to send a NACK (instead of an ACK) after receiving the last byte.<br><br>**NOTE:** Once the SECOND_TO_STS bit in TCO2_STS register (D31:F0, TCOBASE+6h, bit 1) is set, the LAST_BYTE bit also gets set. While the SECOND_TO_STS bit is set, the LAST_BYTE bit cannot be cleared. This prevents the Chipset from running some of the SMBus commands (Block Read/ Write, $I^2C$ Read, Block $I^2C$ Write). |

| Bit | Description |
|---|---|
| 4:2 | **SMB_CMD** — R/W. The bit encoding below indicates which command the Chipset is to perform. If enabled, the Chipset will generate an interrupt or SMI# when the command has completed If the value is for a non-supported or reserved command, the Chipset will set the device error (DEV_ERR) status bit (offset SMBASE + 00h, bit 2) and generate an interrupt when the START bit is set. The Chipset will perform no command, and will not operate until DEV_ERR is cleared.<br>000 = **Quick**: The slave address and read/write value (bit 0) are stored in the transmit slave address register.<br>001 = **Byte**: This command uses the transmit slave address and command registers. Bit 0 of the slave address register determines if this is a read or write command.<br>010 = **Byte Data**: This command uses the transmit slave address, command, and DATA0 registers. Bit 0 of the slave address register determines if this is a read or write command. If it is a read, the DATA0 register will contain the read data.<br>011 = **Word Data**: This command uses the transmit slave address, command, DATA0 and DATA1 registers. Bit 0 of the slave address register determines if this is a read or write command. If it is a read, after the command completes, the DATA0 and DATA1 registers will contain the read data.<br>100 = **Process Call:** This command uses the transmit slave address, command, DATA0 and DATA1 registers. Bit 0 of the slave address register determines if this is a read or write command. After the command completes, the DATA0 and DATA1 registers will contain the read data.<br>101 = **Block**: This command uses the transmit slave address, command, DATA0 registers, and the Block Data Byte register. For block write, the count is stored in the DATA0 register and indicates how many bytes of data will be transferred. For block reads, the count is received and stored in the DATA0 register. Bit 0 of the slave address register selects if this is a read or write command. For writes, data is retrieved from the first n (where n is equal to the specified count) addresses of the SRAM array. For reads, the data is stored in the Block Data Byte register.<br>110 = **$I^2C$ Read**: This command uses the transmit slave address, command, DATA0, DATA1 registers, and the Block Data Byte register. The read data is stored in the Block Data Byte register. The Chipset continues reading data until the NAK is received.<br>**111 = Block Process:** This command uses the transmit slave address, command, DATA0 and the Block Data Byte register. For block write, the count is stored in the DATA0 register and indicates how many bytes of data will be transferred. For block read, the count is received and stored in the DATA0 register. Bit 0 of the slave address register always indicate a write command. For writes, data is retrieved from the first m (where m is equal to the specified count) addresses of the SRAM array. For reads, the data is stored in the Block Data Byte register.<br><br>**NOTE:** E32B bit in the Auxiliary Control register must be set for this command to work. |
| 1 | **KILL** — R/W.<br>0 = Normal SMBus host controller functionality.<br>1 = Kills the current host transaction taking place, sets the FAILED status bit, and asserts the interrupt (or SMI#). This bit, once set, must be cleared by software to allow the SMBus host controller to function normally. |
| 0 | **INTREN** — R/W.<br>0 = Disable.<br>1 = Enable the generation of an interrupt or SMI# upon the completion of the command. |

### 17.2.3 HST_CMD—Host Command Register (SMBUS—D31:F3)

Register Offset: SMBASE + 03h Attribute: R/W
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | This 8-bit field is transmitted by the host controller in the command field of the SMBus protocol during the execution of any command. |

### 17.2.4 XMIT_SLVA—Transmit Slave Address Register (SMBUS—D31:F3)

Register Offset: SMBASE + 04h Attribute: R/W
Default Value: 00h Size: 8 bits

This register is transmitted by the host controller in the slave address field of the SMBus protocol.

| Bit | Description |
|---|---|
| 7:1 | **Address** — R/W. This field provides a 7-bit address of the targeted slave. |
| 0 | **RW** — R/W. Direction of the host transfer.<br>0 = Write<br>1 = Read |

### 17.2.5 HST_D0—Host Data 0 Register (SMBUS—D31:F3)

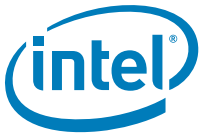
Register Offset: SMBASE + 05h Attribute: R/W
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Data0 / Count** — R/W. This field contains the 8-bit data sent in the DATA0 field of the SMBus protocol. For block write commands, this register reflects the number of bytes to transfer. This register should be programmed to a value between 1 and 32 for block counts. A count of 0 or a count above 32 will result in unpredictable behavior. The host controller does not check or log invalid block counts. |

### 17.2.6 HST_D1—Host Data 1 Register (SMBUS—D31:F3)

Register Offset: SMBASE + 06h Attribute: R/W
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Data1** — R/W. This 8-bit register is transmitted in the DATA1 field of the SMBus protocol during the execution of any command. |



### 17.2.7 Host_BLOCK_DB—Host Block Data Byte Register (SMBUS—D31:F3)

Register Offset: SMBASE + 07h
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Block Data (BDTA)** — R/W. This is either a register, or a pointer into a 32-byte block array, depending upon whether the E32B bit is set in the Auxiliary Control register. When the E32B bit (offset SMBASE + 0Dh, bit 1) is cleared, this is a register containing a byte of data to be sent on a block write or read from on a block read, just as it behaved on the ICH3.<br>When the E32B bit is set, reads and writes to this register are used to access the 32-byte block data storage array. An internal index pointer is used to address the array, which is reset to 0 by reading the HCTL register (offset 02h). The index pointer then increments automatically upon each access to this register. The transfer of block data into (read) or out of (write) this storage array during an SMBus transaction always starts at index address 0.<br>When the E2B bit is set, for writes, software will write up to 32-bytes to this register as part of the setup for the command. After the Host controller has sent the Address, Command, and Byte Count fields, it will send the bytes in the SRAM pointed to by this register.<br>When the E2B bit is cleared for writes, software will place a single byte in this register. After the host controller has sent the address, command, and byte count fields, it will send the byte in this register. If there is more data to send, software will write the next series of bytes to the SRAM pointed to by this register and clear the DONE_STS bit. The controller will then send the next byte. During the time between the last byte being transmitted to the next byte being transmitted, the controller will insert wait-states on the interface.<br>When the E2B bit is set for reads, after receiving the byte count into the Data0 register, the first series of data bytes go into the SRAM pointed to by this register. If the byte count has been exhausted or the 32-byte SRAM has been filled, the controller will generate an SMI# or interrupt (depending on configuration) and set the DONE_STS bit. Software will then read the data. During the time between when the last byte is read from the SRAM to when the DONE_STS bit is cleared, the controller will insert wait-states on the interface. |

### 17.2.8 PEC—Packet Error Check (PEC) Register (SMBUS—D31:F3)

Register Offset: SMBASE + 08h
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **PEC_DATA** — R/W. This 8-bit register is written with the 8-bit CRC value that is used as the SMBus PEC data prior to a write transaction. For read transactions, the PEC data is loaded from the SMBus into this register and is then read by software. Software must ensure that the INUSE_STS bit is properly maintained to avoid having this field over-written by a write transaction following a read transaction. |

### 17.2.9 RCV_SLVA—Receive Slave Address Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | SMBASE + 09h | Attribute: | R/W |
| Default Value: | 44h | Size: | 8 bits |
| Lockable: | No | Power Well: | Resume |

| Bit | Description |
|---|---|
| 7 | **Reserved** |
| 6:0 | **SLAVE_ADDR** — R/W. This field is the slave address that the Chipset decodes for read and write cycles. the default is not 0, so the SMBus Slave Interface can respond even before the processor comes up (or if the processor is dead). This register is cleared by RSMRST#, but not by PLTRST#. |

### 17.2.10 SLV_DATA—Receive Slave Data Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | SMBASE + 0Ah–0Bh | Attribute: | RO |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Resume |

This register contains the 16-bit data value written by the external SMBus master. The processor can then read the value from this register. This register is reset by RSMRST#, but not PLTRST#

| Bit | Description |
|---|---|
| 15:8 | **Data Message Byte 1 (DATA_MSG1)** — RO. See Section 5.21.7 - Volume 1 for a discussion of this field. |
| 7:0 | **Data Message Byte 0 (DATA_MSG0)** — RO. See Section 5.21.7 - Volume 1 for a discussion of this field. |

### 17.2.11 AUX_STS—Auxiliary Status Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | SMBASE + 0Ch | Attribute: | R/WC, RO |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Resume |

| Bit | Description |
|---|---|
| 7:2 | **Reserved** |
| 1 | **SMBus TCO Mode (STCO)** — RO. This bit reflects the strap setting of TCO compatible mode vs. Advanced TCO mode.<br>0 = Chipset is in the compatible TCO mode.<br>1 = Chipset is in the advanced TCO mode. |

| Bit | Description |
|---|---|
| 0 | **CRC Error (CRCE)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit is set if a received message contained a CRC error. When this bit is set, the DERR bit of the host status register will also be set. This bit will be set by the controller if a software abort occurs in the middle of the CRC portion of the cycle or an abort happens after the Chipset has received the final data bit transmitted by an external slave. |

## 17.2.12 AUX_CTL—Auxiliary Control Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | SMBASE + 0Dh | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Resume |

| Bit | Description |
|---|---|
| 7:2 | **Reserved** |
| 1 | **Enable 32-Byte Buffer (E32B)** — R/W.<br>0 = Disable.<br>1 = Enable. When set, the Host Block Data register is a pointer into a 32-byte buffer, as opposed to a single register. This enables the block commands to transfer or receive up to 32-bytes before the Chipset generates an interrupt. |
| 0 | **Automatically Append CRC (AAC)** — R/W.<br>0 = Chipset will Not automatically append the CRC.<br>1 = The Chipset will automatically append the CRC. This bit must not be changed during SMBus transactions or undetermined behavior will result. It should be programmed only once during the lifetime of the function. |

## 17.2.13 SMLINK_PIN_CTL—SMLink Pin Control Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | SMBASE + 0Eh | Attribute: | R/W, RO |
| Default Value: | See below | Size: | 8 bits |

***Note:*** This register is in the resume well and is reset by RSMRST#.

This register is only applicable in the TCO compatible mode.

| Bit | Description |
|---|---|
| 7:3 | **Reserved** |
| 2 | **SMLINK_CLK_CTL** — R/W.<br>0 = Chipset will drive the SMLINK0 pin low, independent of what the other SMLINK logic would otherwise indicate for the SMLINK0 pin.<br>1 = The SMLINK0 pin is **not** overdriven low. The other SMLINK logic controls the state of the pin. (Default) |
| 1 | **SMLINK1_CUR_STS** — RO. This read-only bit has a default value that is dependent on an external signal level. This pin returns the value on the SMLINK1 pin. This allows software to read the current state of the pin.<br>0 = Low<br>1 = High |
| 0 | **SMLINK0_CUR_STS** — RO. This read-only bit has a default value that is dependent on an external signal level. This pin returns the value on the SMLINK0 pin. This allows software to read the current state of the pin.<br>0 = Low<br>1 = High |

## 17.2.14 SMBUS_PIN_CTL—SMBUS Pin Control Register (SMBUS—D31:F3)

Register Offset: SMBASE + 0Fh
Default Value: See below
Attribute: R/W, RO
Size: 8 bits

***Note:*** This register is in the resume well and is reset by RSMRST#.

| Bit | Description |
|---|---|
| 7:3 | **Reserved** |
| 2 | **SMBCLK_CTL** — R/W.<br>1 = The SMBCLK pin is **not** overdriven low. The other SMBus logic controls the state of the pin.<br>0 = Chipset drives the SMBCLK pin low, independent of what the other SMB logic would otherwise indicate for the SMBCLK pin. (Default) |
| 1 | **SMBDATA_CUR_STS** — RO. This read-only bit has a default value that is dependent on an external signal level. This pin returns the value on the SMBDATA pin. This allows software to read the current state of the pin.<br>0 = Low<br>1 = High |
| 0 | **SMBCLK_CUR_STS** — RO. This read-only bit has a default value that is dependent on an external signal level. This pin returns the value on the SMBCLK pin. This allows software to read the current state of the pin.<br>0 = Low<br>1 = High |

## 17.2.15 SLV_STS—Slave Status Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | SMBASE + 10h | Attribute: | R/WC |
| Default Value: | 00h | Size: | 8 bits |

***Note:*** This register is in the resume well and is reset by RSMRST#.

All bits in this register are implemented in the 64 kHz clock domain. Therefore, software must poll this register until a write takes effect before assuming that a write has completed internally.

| Bit | Description |
|---|---|
| 7:1 | **Reserved** |
| 0 | **HOST_NOTIFY_STS** — R/WC. The Chipset sets this bit to a 1 when it has completely received a successful Host Notify Command on the SMLink pins. Software reads this bit to determine that the source of the interrupt or SMI# was the reception of the Host Notify Command. Software clears this bit after reading any information needed from the Notify address and data registers by writing a 1 to this bit. Note that the Chipset will allow the Notify Address and Data registers to be over-written once this bit has been cleared. When this bit is 1, the Chipset will NACK the first byte (host address) of any new “Host Notify” commands on the SMLink. Writing a 0 to this bit has no effect. |

## 17.2.16 SLV_CMD—Slave Command Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | SMBASE + 11h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bits |

***Note:*** This register is in the resume well and is reset by RSMRST#.

| Bit | Description |
|---|---|
| 7:2 | **Reserved** |
| 2 | **SMBALERT_DIS** — R/W.<br>0 = Allows the generation of the interrupt or SMI#.<br>1 = Software sets this bit to block the generation of the interrupt or SMI# due to the SMBALERT# source. This bit is logically inverted and ANDed with the SMBALERT_STS bit (offset SMBASE + 00h, bit 5). The resulting signal is distributed to the SMI# and/or interrupt generation logic. This bit does not effect the wake logic. |
| 1 | **HOST_NOTIFY_WKEN** — R/W. Software sets this bit to 1 to enable the reception of a Host Notify command as a wake event. When enabled this event is “OR”ed in with the other SMBus wake events and is reflected in the SMB_WAK_STS bit of the General Purpose Event 0 Status register.<br>0 = Disable<br>1 = Enable |

| Bit | Description |
|---|---|
| 0 | **HOST_NOTIFY_INTREN** — R/W. Software sets this bit to 1 to enable the generation of interrupt or SMI# when HOST_NOTIFY_STS (offset SMBASE + 10h, bit 0) is 1. This enable does not affect the setting of the HOST_NOTIFY_STS bit. When the interrupt is generated, either PIRQB# or SMI# is generated, depending on the value of the SMB_SMI_EN bit (D31:F3:40h, bit 1). If the HOST_NOTIFY_STS bit is set when this bit is written to a 1, then the interrupt (or SMI#) will be generated. The interrupt (or SMI#) is logically generated by AND'ing the STS and INTREN bits.<br>0 = Disable<br>1 = Enable |

## 17.2.17 NOTIFY_DADDR—Notify Device Address Register (SMBUS—D31:F3)

Register Offset: SMBASE + 14h — Attribute: RO
Default Value: 00h — Size: 8 bits

***Note:*** This register is in the resume well and is reset by RSMRST#.

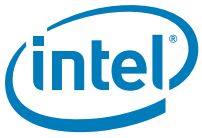
| Bit | Description |
|---|---|
| 7:1 | **DEVICE_ADDRESS** — RO. This field contains the 7-bit device address received during the Host Notify protocol of the SMBus 2.0 Specification. Software should only consider this field valid when the HOST_NOTIFY_STS bit (D31:F3:SMBASE +10, bit 0) is set to 1. |
| 0 | **Reserved** |

## 17.2.18 NOTIFY_DLOW—Notify Data Low Byte Register (SMBUS—D31:F3)

Register Offset: SMBASE + 16h — Attribute: RO
Default Value: 00h — Size: 8 bits

***Note:*** This register is in the resume well and is reset by RSMRST#.

| Bit | Description |
|---|---|
| 7:0 | **DATA_LOW_BYTE** — RO. This field contains the first (low) byte of data received during the Host Notify protocol of the SMBus 2.0 specification. Software should only consider this field valid when the HOST_NOTIFY_STS bit (D31:F3:SMBASE +10, bit 0) is set to 1. |



### 17.2.19 NOTIFY_DHIGH—Notify Data High Byte Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | SMBASE + 17h | Attribute: | RO |
| Default Value: | 00h | Size: | 8 bits |

***Note:*** This register is in the resume well and is reset by RSMRST#.

| Bit | Description |
|---|---|
| 7:0 | **DATA_HIGH_BYTE** — RO. This field contains the second (high) byte of data received during the Host Notify protocol of the SMBus 2.0 specification. Software should only consider this field valid when the HOST_NOTIFY_STS bit (D31:F3:SMBASE +10, bit 0) is set to 1. |

§

# 18 Intel HD Audio Controller Registers (D27:F0)

The Intel HD Audio controller resides in PCI Device 27, Function 0 on bus 0. This function contains a set of DMA engines that are used to move samples of digitally encoded data between system memory and external codecs.

***Note:*** All registers in this function (including memory-mapped registers) must be addressable in byte, word, and D-word quantities. The software must always make register accesses on natural boundaries (i.e. D-word accesses must be on D-word boundaries; word accesses on word boundaries, etc.) In addition, the memory-mapped register space must not be accessed with the LOCK semantic exclusive-access mechanism. If software attempts exclusive-access mechanisms to the Intel High Definition Audio memory-mapped space, the results are undefined.

***Note:*** Users interested in providing feedback on the Intel High Definition Audio specification or planning to implement the Intel High Definition Audio specification into a future product will need to execute the *Intel HD Audio Specification Developer's Agreement*. For more information, contact nextgenaudio@intel.com.

## 18.1 Intel HD Audio PCI Configuration Space (Intel HD Audio— D27:F0)

***Note:*** Address locations that are not shown should be treated as **Reserved**.

**Table 18-146.Intel HD Audio PCI Register Address Map (Intel HD Audio D27:F0) (Sheet 1 of 3)**

| Offset | Mnemonic | Register Name | Default | Access |
|---|---|---|---|---|
| 00h–01h | VID | Vendor Identification | 8086h | RO |
| 02h–03h | DID | Device Identification | See register description. | RO |
| 04h–05h | PCICMD | PCI Command | 0000h | R/W, RO |
| 06h–07h | PCISTS | PCI Status | 0010h | R/WC, RO |
| 08h | RID | Revision Identification | See register description. | RO |
| 09h | PI | Programming Interface | 00h | RO |
| 0Ah | SCC | Sub Class Code | 03h | RO |
| 0Bh | BCC | Base Class Code | 04h | RO |
| 0Ch | CLS | Cache Line Size | 00h | R/W |
| 0Dh | LT | Latency Timer | 00h | RO |
| 0Eh | HEADTYP | Header Type | 00h | RO |
| 10h–13h | HDBARL | Intel HD Audio Lower Base Address (Memory) | 00000004h | R/W, RO |

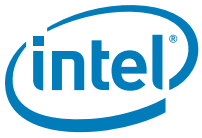


**Table 18-146.Intel HD Audio PCI Register Address Map (Intel HD Audio D27:F0) (Sheet 2 of 3)**

| Offset | Mnemonic | Register Name | Default | Access |
|---|---|---|---|---|
| 14h–17h | HDBARU | Intel HD Audio Upper Base Address (Memory) | 00000000h | R/W |
| 2Ch–2Dh | SVID | Subsystem Vendor Identification | 0000h | R/WO |
| 2Eh–2Fh | SID | Subsystem Identification | 0000h | R/WO |
| 34h | CAPPTR | Capability List Pointer | 50h | RO |
| 3Ch | INTLN | Interrupt Line | 00h | R/W |
| 3Dh | INTPN | Interrupt Pin | See Register Description | RO |
| 40h | HDCTL | Intel High Definition Audio Control | 00h | R/W, RO |
| 44h | TCSEL | Traffic Class Select | 00h | R/W |
| 4C-4Dh | - | **Reserved** | 0000h | RO |
| 50h–51h | PID | PCI Power Management Capability ID | 6001h | RO |
| 52h–53h | PC | Power Management Capabilities | C842 | RO |
| 54h–57h | PCS | Power Management Control and Status | 00000000h | R/W, RO, R/WC |
| 60h–61h | MID | MSI Capability ID | 7005h | RO |
| 62h–63h | MMC | MSI Message Control | 0080h | R/W, RO |
| 64h–67h | MMLA | MSI Message Lower Address | 00000000h | R/W, RO |
| 68h–6Bh | MMUA | SMI Message Upper Address | 00000000h | R/W |
| 6Ch–6Dh | MMD | MSI Message Data | 0000h | R/W |
| 70h–71h | PXID | PCI Express* Capability Identifiers | 0010h | RO |
| 72h–73h | PXC | PCI Express Capabilities | 0091h | RO |
| 74h–77h | DEVCAP | Device Capabilities | 00000000h | RO, R/WO |
| 78h–79h | DEVC | Device Control | 0800h | R/W, RO |
| 7Ah–7Bh | DEVS | Device Status | 0010h | RO |
| 100h–103h | VCCAP | Virtual Channel Enhanced Capability Header | 13010002h | RO |
| 104h–107h | PVCCAP1 | Port VC Capability Register 1 | 00000001h | RO |
| 108h–10Bh | PVCCAP2 | Port VC Capability Register 2 | 00000000h | RO |
| 10Ch–10D | PVCCTL | Port VC Control | 0000h | RO |
| 10Eh–10Fh | PVCSTS | Port VC Status | 0000h | RO |
| 110h–103h | VC0CAP | VC0 Resource Capability | 00000000h | RO |
| 114h–117h | VC0CTL | VC0 Resource Control | 800000FFh | R/W, RO |
| 11Ah–11Bh | VC0STS | VC0 Resource Status | 0000h | RO |
| 11Ch–11Fh | VCiCAP | VCi Resource Capability | 00000000h | RO |
| 120h–123h | VCiCTL | VCi Resource Control | 00000000h | R/W, RO |
| 126h–127h | VCiSTS | VCi Resource Status | 0000h | RO |
| 130h–133h | RCCAP | Root Complex Link Declaration Enhanced Capability Header | 00010005h | RO |
| 134h–137h | ESD | Element Self Description | 0F000100h | RO |

**Table 18-146.Intel HD Audio PCI Register Address Map (Intel HD Audio D27:F0) (Sheet 3 of 3)**

| Offset | Mnemonic | Register Name | Default | Access |
|---|---|---|---|---|
| 140h–143h | L1DESC | Link 1 Description | 00000001h | RO |
| 148h–14Bh | L1ADDL | Link 1 Lower Address | See Register Description | RO |
| 14Ch–14Fh | L1ADDU | Link 1 Upper Address | 00000000h | RO |

## 18.1.1 VID—Vendor Identification Register (Intel HD Audio Controller—D27:F0)

Offset: 00h-01h Attribute: RO

Default Value: 8086hSize:16 bits

| Bit | Description |
|---|---|
| 15:0 | **Vendor ID** — RO. This is a 16-bit value assigned to Intel. Intel VID = 8086h |

## 18.1.2 DID—Device Identification Register (Intel HD Audio Controller—D27:F0)

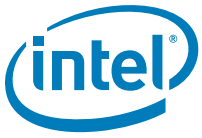
Offset Address: 02h–03h Attribute: RO
Default Value: See bit description Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Device ID** — RO. T |

## 18.1.3 PCICMD—PCI Command Register (Intel HD Audio Controller—D27:F0)

Offset Address: 04h–05h Attribute: R/W, RO
Default Value: 0000h Size: 16 bits

| Bit | Description |
|---|---|
| 15:11 | **Reserved** |
| 10 | **Interrupt Disable (ID)** — R/W.<br>0= The INTx# signals may be asserted.<br>1= The Intel HD Audio controller's INTx# signal will be de-asserted<br><br>**NOTE:** This bit does not affect the generation of MSIs. |
| 9 | **Fast Back to Back Enable (FBE)** — RO. Not implemented. Hardwired to 0. |
| 8 | **SERR# Enable (SERR_EN)** — R/W. SERR# is not generated by the Chipset Intel High Definition Audio Controller. |
| 7 | **Wait Cycle Control (WCC)** — RO. Not implemented. Hardwired to 0. |
| 6 | **Parity Error Response (PER)** — RO. Not implemented. Hardwired to 0. |
| 5 | **VGA Palette Snoop (VPS)**. Not implemented. Hardwired to 0. |



| Bit | Description |
|---|---|
| 4 | **Memory Write and Invalidate Enable** (MWIE) — RO. Not implemented. Hardwired to 0. |
| 3 | Special Cycle Enable (SCE). Not implemented. Hardwired to 0. |
| 2 | **Bus Master Enable (BME)** — R/W. Controls standard PCI Express* bus mastering capabilities for Memory and I/O, reads and writes. Note that this bit also controls MSI generation since MSIs are essentially memory writes.<br>0 = Disable<br>1 = Enable |
| 1 | **Memory Space Enable (MSE)** — R/W. Enables memory space addresses to the Intel High Definition Audio controller.<br>0 = Disable<br>1 = Enable |
| 0 | I/O Space Enable (IOSE)—RO. Hardwired to 0 since the Intel High Definition Audio controller does not implement I/O space. |

## 18.1.4 PCISTS—PCI Status Register (Intel HD Audio Controller—D27:F0)

Offset Address: 06h–07h Attribute: RO, R/WC
Default Value: 0010h Size: 16 bits

| Bit | Description |
|---|---|
| 15 | **Detected Parity Error (DPE)** — RO. Not implemented. Hardwired to 0. |
| 14 | **SERR# Status (SERRS)** — RO. Not implemented. Hardwired to 0. |
| 13 | **Received Master Abort (RMA)** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = No master abort received.<br>1 = The Intel HD Audio controller sets this bit when, as a bus master, it receives a master abort. When set, the Intel High Definition Audio controller clears the run bit for the channel that received the abort. |
| 12 | **Received Target Abort (RTA)** — RO. Not implemented. Hardwired to 0. |
| 11 | **Signaled Target Abort (STA)** — RO. Not implemented. Hardwired to 0. |
| 10:9 | **DEVSEL# Timing Status (DEV_STS)** — RO. Does not apply. Hardwired to 0. |
| 8 | **Data Parity Error Detected (DPED)** — RO. Not implemented. Hardwired to 0. |
| 7 | **Fast Back to Back Capable (FB2BC)** — RO. Does not apply. Hardwired to 0. |
| 6 | **Reserved**. |
| 5 | **66 MHz Capable (66MHZ_CAP)** — RO. Does not apply. Hardwired to 0. |
| 4 | **Capabilities List (CAP_LIST)** — RO. Hardwired to 1. Indicates that the controller contains a capabilities pointer list. The first item is pointed to by looking at configuration offset 34h. |
| 3 | **Interrupt Status (IS)** — RO.<br>0 = This bit is 0 after the interrupt is cleared.<br>1 = This bit is 1 when the INTx# is asserted.<br>Note that this bit is not set by an MSI. |
| 2:0 | **Reserved**. |

### 18.1.5 RID—Revision Identification Register (Intel HD Audio Controller—D27:F0)

Offset: 08h Attribute: RO
Default Value: See bit description Size: 8 Bits

| Bit | Description |
|---|---|
| 7:0 | **Revision ID** — RO. |

### 18.1.6 PI—Programming Interface Register (Intel HD Audio Controller—D27:F0)

Offset: 09h Attribute: RO
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Programming Interface** — RO. |

### 18.1.7 SCC—Sub Class Code Register (Intel HD Audio Controller—D27:F0)

Address Offset: 0Ah Attribute: RO
Default Value: 03h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Sub Class Code (SCC)** — RO.<br>03h = Audio Device |

### 18.1.8 BCC—Base Class Code Register (Intel HD Audio Controller—D27:F0)

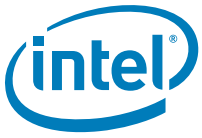
Address Offset: 0Bh Attribute: RO
Default Value: 04h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Base Class Code (BCC)** — RO.<br>04h = Multimedia device |

### 18.1.9 CLS—Cache Line Size Register (Intel HD Audio Controller—D27:F0)

Address Offset: 0Ch Attribute: R/W
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Cache Line Size** — R/W. Implemented as R/W register, but has no functional impact to the Chipset |



### 18.1.10 LT—Latency Timer Register (Intel HD Audio Controller—D27:F0)

Address Offset: 0Dh Attribute: RO
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Latency Timer** — RO. Hardwired to 00 |

### 18.1.11 HEADTYP—Header Type Register (Intel HD Audio Controller—D27:F0)

Address Offset: 0Eh Attribute: RO
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Header Type** — RO. Hardwired to 00. |

### 18.1.12 HDBARL—Intel HD Audio Lower Base Address Register (Intel HD Audio—D27:F0)

Address Offset: 10h-13h Attribute: R/W, RO
Default Value: 00000004h Size: 32 bits

| Bit | Description |
|---|---|
| 31:14 | **Lower Base Address (LBA)** — R/W. This field contains the base address for the Intel HD Audio controller's memory mapped configuration registers; 16 KB are requested by hardwiring bits 13:4 to 0s. |
| 13:4 | RO. Hardwired to 0s |
| 3 | **Prefetchable (PREF)** — RO. Hardwired to 0 to indicate that this BAR is NOT prefetchable. |
| 2:1 | **Address Range (ADDRNG)** — RO. Hardwired to 10b, indicating that this BAR can be located anywhere in 64-bit address space. |
| 0 | **Space Type (SPTYP)** — RO. Hardwired to 0. Indicates this BAR is located in memory space. |

### 18.1.13 HDBARU—Intel HD Audio Upper Base Address Register (Intel HD Audio Controller—D27:F0)

Address Offset: 14h-17h Attribute: R/W
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Upper Base Address (UBA)** — R/W. This field provides the upper 32 bits of the Base address for the Intel HD Audio controller's memory mapped configuration registers. |

### 18.1.14 SVID—Subsystem Vendor Identification Register (Intel HD Audio Controller—D27:F0)

Address Offset: 2Ch–2Dh
Default Value: 0000h
Attribute: R/WO
Size: 16 bits

The SVID register, in combination with the Subsystem ID register (D27:F0:2Eh), enable the operating environment to distinguish one audio subsystem from the other(s).

This register is implemented as write-once register. Once a value is written to it, the value can be read back. Any subsequent writes will have no effect.

This register is not affected by the $D3_{HOT}$ to D0 transition.

| Bit | Description |
|---|---|
| 15:0 | **Subsystem Vendor ID** — R/WO. |

### 18.1.15 SID—Subsystem Identification Register (Intel HD Audio Controller—D27:F0)

Address Offset: 2Eh–2Fh
Default Value: 0000h
Attribute: R/WO
Size: 16 bits

The SID register, in combination with the Subsystem Vendor ID register (D27:F0:2Ch) make it possible for the operating environment to distinguish one audio subsystem from the other(s).

This register is implemented as write-once register. Once a value is written to it, the value can be read back. Any subsequent writes will have no effect.

This register is not affected by the $D3_{HOT}$ to D0 transition.

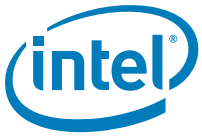
| Bit | Description |
|---|---|
| 15:0 | **Subsystem ID** — R/WO. |

### 18.1.16 CAPPTR—Capabilities Pointer Register (Audio—D30:F2)

Address Offset: 34h
Default Value: 50h
Attribute: RO
Size: 8 bits

This register indicates the offset for the capability pointer.

| Bit | Description |
|---|---|
| 7:0 | **Capabilities Pointer (CAP_PTR)** — RO. This field indicates that the first capability pointer offset is offset 50h (Power Management Capability) |



## 18.1.17 INTLN—Interrupt Line Register (Intel HD Audio Controller—D27:F0)

Address Offset: 3Ch
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Interrupt Line (INT_LN)** — R/W. This data is not used by the Chipset. It is used to communicate to software the interrupt line that the interrupt pin is connected to. |

## 18.1.18 INTPN—Interrupt Pin Register (Intel HD Audio Controller—D27:F0)

Address Offset: 3Dh
Default Value: See Description
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:4 | **Reserved**. |
| 3:0 | **Interrupt Pin** — RO. This reflects the value of D27IP.ZIP (Chipset Config Registers:Offset 3110h: bits 3:0). |

## 18.1.19 HDCTL—Intel HD Audio Control Register (Intel HD Audio Controller—D27:F0)

Address Offset: 40h
Default Value: 00h
Attribute: R/W, RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:4 | **Reserved**. |
| 3 | **BITCLK Detect Clear (CLKDETCLR)** — R/W.<br>0 = lock detect circuit is operational and maybe enabled.<br>1 = Writing a 1 to this bit clears bit 1 (CLKDET#) in this register. CLKDET# bit remains clear when this bit is set to 1.<br><br>**NOTE:** This bit is not affected by the $D3_{HOT}$ to D0 transition. |
| 2 | **BITCLK Detect Enable (CLKDETEN)** — R/W.<br>0 = Latches the current state of bit 1 (CLKDET#) in this register<br>1 = Enables the clock detection circuit<br><br>**NOTE:** This bit is not affected by the $D3_{HOT}$ to D0 transition. |

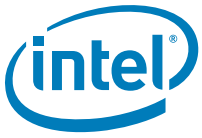
| Bit | Description |
|---|---|
| 1 | **BITCLK Detected Inverted (CLKDET#)** — RO. This bit is modified by hardware.<br>It is set to 0 when the Chipset detects that the BITCLK is toggling, indicating the presence of an AC'97 codec on the link.<br><br>**NOTES:**<br>1. Bit 2 (CLKDETEN) and bit 3 (CLKDETCLR) in this register control the operation of this bit and must be manipulated correctly in order to get a valid CLKDET# indicator.<br>2. This bit is not affected by the D3HOT to D0 transition. |
| 0 | **Intel HD Audio Signal Mode** — R/W. This bit selects the Intel High Definition Audio signals.<br>0 = **Reserved**<br>1 = Intel High Definition Audio mode is selected<br><br>**NOTES:**<br>1. This bit has no effect on the visibility of the Intel High Definition Audio function configuration space.<br>2. This bit is in the resume well and only clear on a power-on reset. Software must not makes assumptions about the reset state of this bit and must set it appropriately. |

## 18.1.20 TCSEL—Traffic Class Select Register (Intel HD Audio Controller—D27:F0)

Address Offset: 44h
Default Value: 00h
Attribute: R/W
Size: 8 bits

This register assigned the value to be placed in the TC field. CORB and RIRB data will always be assigned TC0.

| Bit | Description |
|---|---|
| 7:3 | **Reserved**. |
| 2:0 | **Intel HD Audio Traffic Class Assignment (TCSEL)**— R/W. This register assigns the value to be placed in the Traffic Class field for input data, output data, and buffer descriptor transactions.<br>000 = TC0<br>001 = TC1<br>010 = TC2<br>011 = TC3<br>100 = TC4<br>101 = TC5<br>110 = TC6<br>111 = TC7<br>**NOTE:** These bits are not reset on $D3_{HOT}$ to D0 transition; however, they are reset by PLTRST#. |



### 18.1.21 PID—PCI Power Management Capability ID Register (Intel HD Audio Controller—D27:F0)

Address Offset: 50h–51h Attribute: RO
Default Value: 6001h Size: 16 bits

| Bit | Description |
|---|---|
| 15:8 | **Next Capability (Next)** — RO. Hardwired to 60h. Points to the next capability structure (MSI) |
| 7:0 | **Cap ID (CAP)** — RO. Hardwired to 01h. Indicates that this pointer is a PCI power management capability. |

### 18.1.22 PC—Power Management Capabilities Register (Intel HD Audio Controller—D27:F0)

Address Offset: 52h–53h Attribute: RO
Default Value: C842h Size: 16 bits

| Bit | Description |
|---|---|
| 15:11 | **PME Support** — RO. Hardwired to 11001b. Indicates PME# can be generated from D3 and D0 states. |
| 10 | **D2 Support** — RO. Hardwired to 0. Indicates that D2 state is not supported. |
| 9 | **D1 Support** —RO. Hardwired to 0. Indicates that D1 state is not supported. |
| 8:6 | **Aux Current** — RO. Hardwired to 001b. Reports 55 mA maximum suspend well current required when in the $D3_{COLD}$ state. |
| 5 | **Device Specific Initialization (DSI)** — RO. Hardwired to 0. Indicates that no device specific initialization is required. |
| 4 | **Reserved** |
| 3 | **PME Clock (PMEC)** — RO. Does not apply. Hardwired to 0. |
| 2:0 | **Version** — RO. Hardwired to 010b. Indicates support for version 1.1 of the PCI Power Management Specification. |

### 18.1.23 PCS—Power Management Control and Status Register (Intel HD Audio Controller—D27:F0)

Address Offset: 54h–57h Attribute: RO, R/W, R/WC
Default Value: 00000000h Size: 32 bits

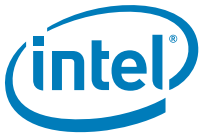
| Bit | Description |
|---|---|
| 31:24 | **Data** — RO. Does not apply. Hardwired to 0. |
| 23 | **Bus Power/Clock Control Enable** — RO. Does not apply. Hardwired to 0. |
| 22 | **B2/B3 Support** — RO. Does not apply. Hardwired to 0. |
| 21:16 | **Reserved**. |

| Bit | Description |
|---|---|
| 15 | **PME Status (PMES)** — R/WC.<br>0 = Software clears the bit by writing a 1 to it.<br>1 = This bit is set when the Intel HD Audio controller would normally assert the PME# signal independent of the state of the PME_EN bit (bit 8 in this register)<br>This bit is in the resume well and only cleared on a power-on reset. Software must not make assumptions about the reset state of this bit and must set it appropriately. |
| 14:9 | **Reserved** |
| 8 | **PME Enable (PMEE)** — R/W.<br>0 = Disable<br>1 = when set and if corresponding PMES also set, the Intel High Definition Audio controller sets the AC97_STS bit in the GPE0_STS register (PMBASE +28h).<br>This bit is in the resume well and only cleared on a power-on reset. Software must not make assumptions about the reset state of this bit and must set it appropriately. |
| 7:2 | **Reserved** |
| 1:0 | **Power State (PS)** — R/W. This field is used both to determine the current power state of the Intel High Definition Audio controller and to set a new power state.<br>00 = D0 state<br>11 = $D3_{HOT}$ state<br>Others = reserved<br><br>**NOTES:**<br>1. If software attempts to write a value of 01b or 10b in to this field, the write operation must complete normally; however, the data is discarded and no state change occurs.<br>2. When in the $D3_{HOT}$ states, the Intel High Definition Audio controller's configuration space is available, but the I/O and memory space are not. Additionally, interrupts are blocked.<br>3. When software changes this value from $D3_{HOT}$ state to the D0 state, an internal warm (soft) reset is generated, and software must re-initialize the function. |

## 18.1.24 MID—MSI Capability ID Register (Intel HD Audio Controller—D27:F0)

Address Offset: 60h–61h
Default Value: 7005h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:8 | **Next Capability (Next)** — RO. Hardwired to 70h. Points to the PCI Express* capability structure. |
| 7:0 | **Cap ID (CAP)** — RO. Hardwired to 05h. Indicates that this pointer is a MSI capability |



## 18.1.25 MMC—MSI Message Control Register (Intel HD Audio Controller—D27:F0)

Address Offset: 62h–63h Attribute: RO, R/W
Default Value: 0080h Size: 16 bits

| Bit | Description |
|---|---|
| 15:8 | **Reserved** |
| 7 | **64b Address Capability (64ADD)** — RO. Hardwired to 1 indicating the ability to generate a 64-bit message address |
| 6:4 | **Multiple Message Enable (MME)** — RO. Normally this is a R/W register. However, since only 1 message is supported, these bits are hardwired to 000 = 1 message. |
| 3:1 | **Multiple Message Capable (MMC)** — RO. Hardwired to 0 indicating request for 1 message. |
| 0 | **MSI Enable (ME)** — R/W.<br>0 = an MSI may not be generated<br>1 = an MSI will be generated instead of an INTx signal. |

## 18.1.26 MMLA—MSI Message Lower Address Register (Intel HD Audio Controller—D27:F0)

Address Offset: 64h–67h Attribute: RO, R/W
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:2 | **Message Lower Address (MLA)** — R/W. Lower address used for MSI message. |
| 1:0 | **Reserved**. |

## 18.1.27 MMUA—MSI Message Upper Address Register (Intel HD Audio Controller—D27:F0)

Address Offset: 68h–6Bh Attribute: R/W
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Message Upper Address (MUA)** — R/W. Upper 32-bits of address used for MSI message. |

## 18.1.28 MMD—MSI Message Data Register (Intel HD Audio Controller—D27:F0)

Address Offset: 6Ch–6Dh Attribute: R/W
Default Value: 0000h Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Message Data (MD)** — R/W. Data used for MSI message. |

### 18.1.29 PXID—PCI Express* Capability ID Register (Intel HD Audio Controller—D27:F0)

Address Offset: 70h-71h
Default Value: 0010h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:8 | **Next Capability (Next)** — RO. Hardwired to 0. Indicates that this is the last capability structure in the list. |
| 7:0 | **Cap ID (CAP)** — RO. Hardwired to 10h. Indicates that this pointer is a PCI Express* capability structure |

### 18.1.30 PXC—PCI Express* Capabilities Register (Intel HD Audio Controller—D27:F0)

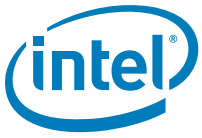
Address Offset: 72h–73h
Default Value: 0091h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:14 | **Reserved** |
| 13:9 | **Interrupt Message Number (IMN)** — RO. Hardwired to 0. |
| 8 | **Slot Implemented (SI)** — RO. Hardwired to 0. |
| 7:4 | **Device/Port Type (DPT)** — RO. Hardwired to 1001b. Indicates that this is a Root Complex Integrated endpoint device. |
| 3:0 | **Capability Version (CV)** — RO. Hardwired to 0001b. Indicates version #1 PCI Express capability |

### 18.1.31 DEVCAP—Device Capabilities Register (Intel HD Audio Controller—D27:F0)

Address Offset: 74h–77h
Default Value: 00000000h
Attribute: R/WO, RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:28 | **Reserved** |
| 27:26 | **Captured Slot Power Limit Scale (SPLS)** — RO. Hardwired to 0. |
| 25:18 | **Captured Slot Power Limit Value (SPLV)** — RO. Hardwired to 0. |
| 17:15 | **Reserved** |
| 14 | **Power Indicator Present** — RO. Hardwired to 0. |
| 13 | **Attention Indicator Present** — RO. Hardwired to 0. |
| 12 | **Attention Button Present** — RO. Hardwired to 0. |
| 11:9 | **Endpoint L1 Acceptable Latency** — R/WO. |
| 8:6 | **Endpoint L0s Acceptable Latency** — R/WO. |
| 5 | **Extended Tag Field Support** — RO. Hardwired to 0. Indicates 5-bit tag field support |
| 4:3 | **Phantom Functions Supported** — RO. Hardwired to 0. Indicates that phantom functions are not supported |



| Bit | Description |
|---|---|
| 2:0 | **Max Payload Size Supported** — RO. Hardwired to 0. Indicates 128-B maximum payload size capability |

## 18.1.32 DEVC—Device Control Register (Intel HD Audio Controller—D27:F0)

Address Offset: 78h–79h Attribute: R/W, RO
Default Value: 0800h Size: 16 bits

| Bit | Description |
|---|---|
| 15 | **Reserved** |
| 14:12 | **Max Read Request Size** — RO. Hardwired to 0 enabling 128B maximum read request size. |
| 11 | **No Snoop Enable (NSNPEN)** — R/W.<br>0 = The Intel HD Audio controller will not set the No Snoop bit. In this case, isochronous transfers will not use VC1 (VCi) even if it is enabled since VC1 is not snooped. Isochronous transfers will use VC0.<br>1 = The Intel High Definition Audio controller is permitted to set the No Snoop bit in the Requester Attributes of a bus master transaction. In this case, VC0 or VC1 may be used for isochronous transfers.<br>Note: This bit is not reset on $D3_{HOT}$ to D0 transition; however, it is reset by PLTRST#. |
| 10 | **Auxiliary Power Enable** — RO. Hardwired to 0, indicating that Intel High Definition Audio device does not draw AUX power |
| 9 | **Phantom Function Enable** — RO. Hardwired to 0 disabling phantom functions. |
| 8 | **Extended Tag Field Enable** — RO. Hardwired to 0 enabling 5-bit tag. |
| 7:5 | **Max Payload Size** — RO. Hardwired to 0 indicating 128B. |
| 4 | **Enable Relaxed Ordering** — RO. Hardwired to 0 disabling relaxed ordering. |
| 3 | **Unsupported Request Reporting Enable** — RO. Not implemented. Hardwired to 0. |
| 2 | **Fatal Error Reporting Enable** — RO. Not implemented. Hardwired to 0. |
| 1 | **Non-Fatal Error Reporting Enable** — RO. Not implemented. Hardwired to 0. |
| 0 | **Correctable Error Reporting Enable** — RO. Not implemented. Hardwired to 0. |

## 18.1.33 DEVS—Device Status Register (Intel HD Audio Controller—D27:F0)

Address Offset: 7Ah–7Bh Attribute: RO
Default Value: 0010h Size: 16 bits

| Bit | Description |
|---|---|
| 15:6 | **Reserved** |
| 5 | **Transactions Pending** — RO.<br>0 = Indicates that completions for all non-posted requests have been received.<br>1 = Indicates that Intel HD Audio controller has issued non-posted requests that have not been completed. |
| 4 | **AUX Power Detected** — RO. Hardwired to 1 indicating the device is connected to resume power. |

| Bit | Description |
|---|---|
| 3 | **Unsupported Request Detected** — RO. Not implemented. Hardwired to 0. |
| 2 | **Fatal Error Detected** — RO. Not implemented. Hardwired to 0. |
| 1 | **Non-Fatal Error Detected** — RO. Not implemented. Hardwired to 0. |
| 0 | **Correctable Error Detected** — RO. Not implemented. Hardwired to 0. |

## 18.1.34 VCCAP—Virtual Channel Enhanced Capability Header (Intel HD Audio Controller—D27:F0)

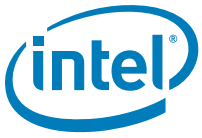
Address Offset: 100h–103h Attribute: RO
Default Value: 13010002h Size: 32 bits

| Bit | Description |
|---|---|
| 31:20 | **Next Capability Offset** — RO. Hardwired to 130h. Points to the next capability header that is the Root Complex Link Declaration Enhanced Capability Header. |
| 19:16 | **Capability Version** — RO. Hardwired to 1h. |
| 15:0 | **PCI Express* Extended Capability** — RO. Hardwired to 0002h. |

## 18.1.35 PVCCAP1—Port VC Capability Register 1 (Intel HD Audio Controller—D27:F0)

Address Offset: 104h–107h Attribute: RO
Default Value: 00000001h Size: 32 bits

| Bit | Description |
|---|---|
| 31:12 | **Reserved**. |
| 11:10 | **Port Arbitration Table Entry Size** — RO. Hardwired to 0 since this is an endpoint device. |
| 9:8 | **Reference Clock** — RO. Hardwired to 0 since this is an endpoint device. |
| 7 | **Reserved**. |
| 6:4 | **Low Priority Extended VC Count** — RO. Hardwired to 0. Indicates that only VC0 belongs to the low priority VC group. |
| 3 | **Reserved**. |
| 2:0 | **Extended VC Count** — RO. Hardwired to 001b. Indicates that 1 extended VC (in addition to VC0) is supported by the Intel HD Audio controller. |



## 18.1.36 PVCCAP2 — Port VC Capability Register 2 (Intel HD Audio Controller—D27:F0)

Address Offset: 108h–10Bh Attribute: RO
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:24 | **VC Arbitration Table Offset** — RO. Hardwired to 0 indicating that a VC arbitration table is not present. |
| 23:8 | **Reserved**. |
| 7:0 | **VC Arbitration Capability** — RO. Hardwired to 0. These bits are not applicable since the Intel HD Audio controller reports a 0 in the Low Priority Extended VC Count bits in the PVCCAP1 register. |

## 18.1.37 PVCCTL — Port VC Control Register (Intel HD Audio Controller—D27:F0)

Address Offset: 10Ch–10Dh Attribute: RO
Default Value: 0000h Size: 16 bits

| Bit | Description |
|---|---|
| 15:4 | **Reserved**. |
| 3:1 | **VC Arbitration Select** — RO. Hardwired to 0. Normally these bits are R/W. However, these bits are not applicable since the Intel HD Audio controller reports a 0 in the Low Priority Extended VC Count bits in the PVCCAP1 register. |
| 0 | **Load VC Arbitration Table** — RO. Hardwired to 0 since an arbitration table is not present. |

## 18.1.38 PVCSTS—Port VC Status Register (Intel HD Audio Controller—D27:F0)

Address Offset: 10Eh-10Fh Attribute: RO
Default Value: 0000h Size: 16 bits

| Bit | Description |
|---|---|
| 15:1 | **Reserved**. |
| 0 | **VC Arbitration Table Status** — RO. Hardwired to 0 since an arbitration table is not present. |

### 18.1.39 VC0CAP—VC0 Resource Capability Register (Intel HD Audio Controller—D27:F0)

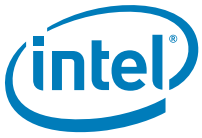
Address Offset: 110h–113h Attribute: RO
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:24 | **Port Arbitration Table Offset** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |
| 23 | **Reserved**. |
| 22:16 | **Maximum Time Slots** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |
| 15 | **Reject Snoop Transactions** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |
| 14 | **Advanced Packet Switching** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |
| 13:8 | **Reserved**. |
| 7:0 | **Port Arbitration Capability** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |

### 18.1.40 VC0CTL—VC0 Resource Control Register (Intel HD Audio Controller—D27:F0)

Address Offset: 114h–117h Attribute: R/W, RO
Default Value: 800000FFh Size: 32 bits

| Bit | Description |
|---|---|
| 31 | **VC0 Enable** — RO. Hardwired to 1 for VC0. |
| 30:27 | **Reserved**. |
| 26:24 | **VC0 ID — RO.** Hardwired to 0 since the first VC is always assigned as VC0. |
| 23:20 | **Reserved**. |
| 19:17 | **Port Arbitration Select** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |
| 16 | **Load Port Arbitration Table** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |
| 15:8 | **Reserved**. |
| 7:0 | **TC/VC0 Map** — R/W, RO. Bit 0 is hardwired to 1 since TC0 is always mapped VC0. Bits [7:1] are implemented as R/W bits. |



## 18.1.41 VC0STS—VC0 Resource Status Register (Intel HD Audio Controller—D27:F0)

Address Offset: 11Ah–11Bh Attribute: RO
Default Value: 0000h Size: 16 bits

| Bit | Description |
|---|---|
| 15:2 | **Reserved**. |
| 1 | **VC0 Negotiation Pending** — RO. Hardwired to 0 since this bit does not apply to the integrated Intel HD Audio device. |
| 0 | **Port Arbitration Table Status** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |

## 18.1.42 VCiCAP—VCi Resource Capability Register (Intel HD Audio Controller—D27:F0)

Address Offset: 11Ch–11Fh Attribute: RO
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:24 | **Port Arbitration Table Offset** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |
| 23 | **Reserved**. |
| 22:16 | **Maximum Time Slots** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |
| 15 | **Reject Snoop Transactions** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |
| 14 | **Advanced Packet Switching** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |
| 13:8 | **Reserved** |
| 7:0 | **Port Arbitration Capability** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |

## 18.1.43 VCiCTL—VCi Resource Control Register (Intel HD Audio Controller—D27:F0)

Address Offset: 120h–123h Attribute: R/W, RO
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31 | **VCi Enable** — R/W.<br>0 = VCi is disabled<br>1 = VCi is enabled<br><br>**NOTE:** This bit is not reset on $D3_{HOT}$ to D0 transition; however, it is reset by PLTRST#. |
| 30:27 | **Reserved**. |
| 26:24 | **VCi ID** — R/W. This field assigns a VC ID to the VCi resource. This field is not used by the Chipset hardware, but it is R/W to avoid confusing software. |

| Bit | Description |
|---|---|
| 23:20 | **Reserved**. |
| 19:17 | **Port Arbitration Select** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |
| 16 | **Load Port Arbitration Table** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |
| 15:8 | **Reserved**. |
| 7:0 | **TC/VCi Map** — R/W, RO. This field indicates the TCs that are mapped to the VCi resource. Bit 0 is hardwired to 0 indicating that it cannot be mapped to VCi. Bits [7:1] are implemented as R/W bits. This field is not used by the Chipset hardware, but it is R/W to avoid confusing software. |

## 18.1.44 VCiSTS—VCi Resource Status Register (Intel HD Audio Controller—D27:F0)

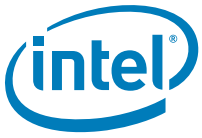
Address Offset: 126h–127h Attribute: RO
Default Value: 0000h Size: 16 bits

| Bit | Description |
|---|---|
| 15:2 | **Reserved**. |
| 1 | **VCi Negotiation Pending** — RO. Does not apply. Hardwired to 0. |
| 0 | **Port Arbitration Table Status** — RO. Hardwired to 0 since this field is not valid for endpoint devices. |

## 18.1.45 RCCAP—Root Complex Link Declaration Enhanced Capability Header Register (Intel HD Audio Controller—D27:F0)

Address Offset: 130h–133h Attribute: RO
Default Value: 00010005h Size: 32 bits

| Bit | Description |
|---|---|
| 31:20 | **Next Capability Offset** — RO. Hardwired to 0 indicating this is the last capability. |
| 19:16 | **Capability Version** — RO. Hardwired to 1h. |
| 15:0 | **PCI Express* Extended Capability ID** — RO. Hardwired to 0005h. |



## 18.1.46 ESD—Element Self Description Register (Intel HD Audio Controller—D27:F0)

Address Offset: 134h–137h Attribute: RO
Default Value: 0F000100h Size: 32 bits

| Bit | Description |
|---|---|
| 31:24 | **Port Number** — RO. Hardwired to 0Fh indicating that the Intel HD Audio controller is assigned as Port #15d. |
| 23:16 | **Component ID** — RO. This field returns the value of the ESD.CID field of the chip configuration section. ESD.CID is programmed by BIOS. |
| 15:8 | **Number of Link Entries** — RO. The Intel High Definition Audio only connects to one device, the Chipset egress port. Therefore this field reports a value of 1h. |
| 7:4 | **Reserved**. |
| 3:0 | **Element Type (ELTYP)** — RO. The Intel High Definition Audio controller is an integrated Root Complex Device. Therefore, the field reports a value of 0h. |

## 18.1.47 L1DESC—Link 1 Description Register (Intel HD Audio Controller—D27:F0)

Address Offset: 140h–143h Attribute: RO
Default Value: 00000001h Size: 32 bits

| Bit | Description |
|---|---|
| 31:24 | **Target Port Number** — RO. The Intel High Definition Audio controller targets the chipset's Port #0. |
| 23:16 | **Target Component ID** — RO. This field returns the value of the ESD.CID field of the chip configuration section. ESD.CID is programmed by BIOS. |
| 15:2 | **Reserved**. |
| 1 | **Link Type** — RO. Hardwired to 0 indicating Type 0. |
| 0 | **Link Valid** — RO. Hardwired to 1. |

## 18.1.48 L1ADDL—Link 1 Lower Address Register (Intel HD Audio Controller—D27:F0)

Address Offset: 148h–14Bh Attribute: RO
Default Value: See Register Description Size: 32 bits

| Bit | Description |
|---|---|
| 31:14 | **Link 1 Lower Address** — RO. Hardwired to match the RCBA register value in the PCI-LPC bridge (D31:F0:F0h). |
| 13:0 | **Reserved**. |

### 18.1.49 L1ADDU—Link 1 Upper Address Register (Intel HD Audio Controller—D27:F0)

Address Offset: 14Ch–14Fh
Default Value: 00000000h
Attribute: RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Link 1 Upper Address** — RO. Hardwired to 00000000h. |

## 18.2 Intel HD Audio Memory-Mapped Configuration Registers (Intel HD Audio— D27:F0)

The base memory location for these memory mapped configuration registers is specified in the HDBAR register (D27:F0:offset 10h and D27:F0:offset 14h). The individual registers are then accessible at HDBAR + Offset as indicated in the following table.

These memory mapped registers must be accessed in byte, word, or DWord quantities.

**Table 18-147.Intel HD Audio PCI Register Address Map (Intel HD Audio D27:F0) (Sheet 1 of 5)**

| HDBAR + Offset | Mnemonic | Register Name | Default | Access |
|---|---|---|---|---|
| 00h–01h | GCAP | Global Capabilities | 4401h | RO |
| 02h | VMIN | Minor Version | 00h | RO |
| 03h | VMAJ | Major Version | 01h | RO |
| 04h–05h | OUTPAY | Output Payload Capability | 003Ch | RO |
| 06h–07h | INPAY | Input Payload Capability | 001Dh | RO |
| 08h–0Bh | GCTL | Global Control | 00000000h | R/W |
| 0Ch–0Dh | WAKEEN | Wake Enable | 0000h | R/W |
| 0Eh–0Fh | STATESTS | State Change Status | 0000h | R/WC |
| 10h–11h | GSTS | Global Status | 0000h | R/WC |
| 12h–13h | — | **Reserved** | 0000h | RO |
| 14h–17h | — | **Reserved** | 00000000h | RO |
| 1 8h–19h | OUTSTRMPAY | Output Stream Payload Capability | 0030h | RO |
| 1Ah–1Bh | INSTRMPAY | Input Stream Payload Capability | 0018h | RO |
| 1Ch–1Fh | — | **Reserved** | 00000000h | RO |
| 20h–23h | INTCTL | Interrupt Control | 00000000h | R/W |
| 24h–27h | INTSTS | Interrupt Status | 00000000h | RO |
| 30h–33h | WALCLK | Wall Clock Counter | 00000000h | RO |
| 34h–37h | SSYNC | Stream Synchronization | 00000000h | R/W |
| 40h–43h | CORBLBASE | CORB Lower Base Address | 00000000h | R/W, RO |

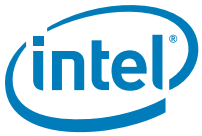


**Table 18-147.Intel HD Audio PCI Register Address Map (Intel HD Audio D27:F0) (Sheet 2 of 5)**

| HDBAR + Offset | Mnemonic | Register Name | Default | Access |
|---|---|---|---|---|
| 44h–47h | CORBUBASE | CORB Upper Base Address | 00000000h | R/W |
| 48h–49h | CORBWP | CORB Write Pointer | 0000h | R/W |
| 4Ah–4Bh | CORBRP | CORB Read Pointer | 0000h | R/W |
| 4Ch | CORBCTL | CORB Control | 00h | R/W |
| 4Dh | CORBST | CORB Status | 00h | R/WC |
| 4Eh | CORBSIZE | CORB Size | 42h | RO |
| 50h–53h | RIRBLBASE | RIRB Lower Base Address | 00000000h | R/W, RO |
| 54h–57h | RIRBUBASE | RIRB Upper Base Address | 00000000h | R/W |
| 58h–59h | RIRBWP | RIRB Write Pointer | 0000h | R/W, RO |
| 5Ah–5Bh | RINTCNT | Response Interrupt Count | 0000h | R/W |
| 5Ch | RIRBCTL | RIRB Control | 00h | R/W |
| 5Dh | RIRBSTS | RIRB Status | 00h | R/WC |
| 5Eh | RIRBSIZE | RIRB Size | 42h | RO |
| 60h–63h | IC | Immediate Command | 00000000h | R/W |
| 64h–67h | IR | Immediate Response | 00000000h | RO |
| 68h–69h | IRS | Immediate Command Status | 0000h | R/W, R/WC |
| 70h–73h | DPLBASE | DMA Position Lower Base Address | 00000000h | R/W, RO |
| 74h–77h | DPUBASE | DMA Position Upper Base Address | 00000000h | R/W |
| 80–82h | ISD0CTL | Input Stream Descriptor 0 (ISD0) Control | 040000h | R/W, RO |
| 83h | ISD0STS | ISD0 Status | 00h | R/WC, RO |
| 84h–87h | ISD0LPIB | ISD0 Link Position in Buffer | 00000000h | RO |
| 88h–8Bh | ISD0CBL | ISD0 Cyclic Buffer Length | 00000000h | R/W |
| 8Ch–8Dh | ISD0LVI | ISD0 Last Valid Index | 0000h | R/W |
| 8Eh–8F | ISD0FIFOW | ISD0 FIFO Watermark | 0004h | R/W |
| 90h–91h | ISD0FIFOS | ISD0 FIFO Size | 0077h | RO |
| 92h–93h | ISD0FMT | ISD0 Format | 0000h | R/W |
| 98h–9Bh | ISD0BDPL | ISD0 Buffer Descriptor List Pointer-Lower Base Address | 00000000h | R/W, RO |
| 9Ch–9Fh | ISD0BDPU | ISD0 Buffer Description List Pointer-Upper Base Address | 00000000h | R/W |
| A0h–A2h | ISD1CTL | Input Stream Descriptor 1(ISD01) Control | 040000h | R/W, RO |
| A3h | ISD1STS | ISD1 Status | 00h | R/WC, RO |
| A4h–A7h | ISD1LPIB | ISD1 Link Position in Buffer | 00000000h | RO |
| A8h–ABh | ISD1CBL | ISD1 Cyclic Buffer Length | 00000000h | R/W |

**Table 18-147.Intel HD Audio PCI Register Address Map (Intel HD Audio D27:F0) (Sheet 3 of 5)**

| HDBAR + Offset | Mnemonic | Register Name | Default | Access |
|---|---|---|---|---|
| ACh–ADh | ISD1LVI | ISD1 Last Valid Index | 0000h | R/W |
| AEh–AFh | ISD1FIFOW | ISD1 FIFO Watermark | 0004h | R/W |
| B0h–B1h | ISD1FIFOS | ISD1 FIFO Size | 0077h | RO |
| B2–B3h | ISD1FMT | ISD1 Format | 0000h | R/W |
| B8–BBh | ISD1BDPL | ISD1 Buffer Descriptor List Pointer-Lower Base Address | 00000000h | R/W, RO |
| BCh–BFh | ISD1BDPU | ISD1 Buffer Description List Pointer-Upper Base Address | 00000000h | R/W |
| C0h–C2h | ISD2CTL | Input Stream Descriptor 2 (ISD2) Control | 040000h | R/W, RO |
| C3h | ISD2STS | ISD2 Status | 00h | R/WC, RO |
| Ch4–C7h | ISD2LPIB | ISD2 Link Position in Buffer | 00000000h | RO |
| C8h–CBh | ISD2CBL | ISD2 Cyclic Buffer Length | 00000000h | R/W |
| CCh–CDh | ISD2LVI | ISD2 Last Valid Index | 0000h | R/W |
| CEh–CFh | ISD1FIFOW | ISD1 FIFO Watermark | 0004h | R/W |
| D0h–D1h | ISD2FIFOS | ISD2 FIFO Size | 0077h | RO |
| D2h–D3h | ISD2FMT | ISD2 Format | 0000h | R/W |
| D8h–DBh | ISD2BDPL | ISD2 Buffer Descriptor List Pointer-Lower Base Address | 00000000h | R/W, RO |
| DCh–DFh | ISD2BDPU | ISD2 Buffer Description List Pointer-Upper Base Address | 00000000h | R/W |
| E0h–E2h | ISD3CTL | Input Stream Descriptor 3 (ISD3) Control | 040000h | R/W, RO |
| E3h | ISD3STS | ISD3 Status | 00h | R/WC, RO |
| E4h–E7h | ISD3LPIB | ISD3 Link Position in Buffer | 00000000h | RO |
| E8h–EBh | ISD3CBL | ISD3 Cyclic Buffer Length | 00000000h | R/W |
| ECh–EDh | ISD3LVI | ISD3 Last Valid Index | 0000h | R/W |
| EEh–EFh | ISD3FIFOW | ISD3 FIFO Watermark | 0004h | R/W |
| F0h–F1h | ISD3FIFOS | ISD3 FIFO Size | 0077h | RO |
| F2h–F3h | ISD3FMT | ISD3 Format | 0000h | R/W |
| F8h–FBh | ISD3BDPL | ISD3 Buffer Descriptor List Pointer-Lower Base Address | 00000000h | R/W, RO |
| FCh–FFh | ISD3BDPU | ISD3 Buffer Description List Pointer-Upper Base Address | 00000000h | R/W |
| 100h–102h | OSD0CTL | Output Stream Descriptor 0 (OSD0) Control | 040000h | R/W, RO |
| 103h | OSD0STS | OSD0 Status | 00h | R/WC, RO |
| 104h–107h | OSD0LPIB | OSD0 Link Position in Buffer | 00000000h | RO |

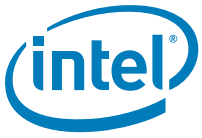


**Table 18-147.Intel HD Audio PCI Register Address Map (Intel HD Audio D27:F0) (Sheet 4 of 5)**

| HDBAR + Offset | Mnemonic | Register Name | Default | Access |
|---|---|---|---|---|
| 108h–10Bh | OSD0CBL | OSD0 Cyclic Buffer Length | 00000000h | R/W |
| 10Ch–10Dh | OSD0LVI | OSD0 Last Valid Index | 0000h | R/W |
| 10Eh–10Fh | OSD0FIFOW | OSD0 FIFO Watermark | 0004h | R/W |
| 110h–111h | OSD0FIFOS | OSD0 FIFO Size | 00BFh | R/W |
| 112–113h | OSD0FMT | OSD0 Format | 0000h | R/W |
| 118h–11Bh | OSD0BDPL | OSD0 Buffer Descriptor List Pointer-Lower Base Address | 00000000h | R/W, RO |
| 11Ch–11Fh | OSD0BDPU | OSD0 Buffer Description List Pointer-Upper Base Address | 00000000h | R/W |
| 120h–122h | OSD1CTL | Output Stream Descriptor 1 (OSD1) Control | 040000h | R/W, RO |
| 123h | OSD1STS | OSD1 Status | 00h | R/WC, RO |
| 124h–127h | OSD1LPIB | OSD1 Link Position in Buffer | 00000000h | RO |
| 128h–12Bh | OSD1CBL | OSD1 Cyclic Buffer Length | 00000000h | R/W |
| 12Ch–12Dh | OSD1LVI | OSD1 Last Valid Index | 0000h | R/W |
| 12Eh–12Fh | OSD1FIFOW | OSD1 FIFO Watermark | 0004h | R/W |
| 130h–131h | OSD1FIFOS | OSD1 FIFO Size | 00BFh | R/W |
| 132h–133h | OSD1FMT | OSD1 Format | 0000h | R/W |
| 138h–13Bh | OSD1BDPL | OSD1 Buffer Descriptor List Pointer-Lower Base Address | 00000000h | R/W, RO |
| 13Ch–13Fh | OSD1BDPU | OSD1 Buffer Description List Pointer-Upper Base Address | 00000000h | R/W |
| 140h–142h | OSD2CTL | Output Stream Descriptor 2 (OSD2) Control | 040000h | R/W, RO |
| 143h | OSD2STS | OSD2 Status | 00h | R/WC, RO |
| 144h–147h | OSD2LPIB | OSD2 Link Position in Buffer | 00000000h | RO |
| 148h–14Bh | OSD2CBL | OSD2 Cyclic Buffer Length | 00000000h | R/W |
| 14Ch–14Dh | OSD2LVI | OSD2 Last Valid Index | 0000h | R/W |
| 14Eh–14Fh | OSD2FIFOW | OSD2 FIFO Watermark | 0004h | R/W |
| 150h–151h | OSD2FIFOS | OSD2 FIFO Size | 00BFh | R/W |
| 152h–153h | OSD2FMT | OSD2 Format | 0000h | R/W |
| 158h–15Bh | OSD2BDPL | OSD2 Buffer Descriptor List Pointer-Lower Base Address | 00000000h | R/W, RO |
| 15Ch–15Fh | OSD2BDPU | OSD2 Buffer Description List Pointer-Upper Base Address | 00000000h | R/W |
| 160h–162h | OSD3CTL | Output Stream Descriptor 3 (OSD3) Control | 040000h | R/W, RO |
| 163h | OSD3STS | OSD3 Status | 00h | R/WC, RO |

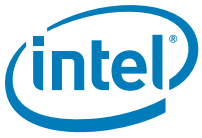
**Table 18-147.Intel HD Audio PCI Register Address Map (Intel HD Audio D27:F0) (Sheet 5 of 5)**

| HDBAR + Offset | Mnemonic | Register Name | Default | Access |
|---|---|---|---|---|
| 164h–167h | OSD3LPIB | OSD3 Link Position in Buffer | 00000000h | RO |
| 168h–16Bh | OSD3CBL | OSD3 Cyclic Buffer Length | 00000000h | R/W |
| 16Ch–16Dh | OSD3LVI | OSD3 Last Valid Index | 0000h | R/W |
| 16Eh–16Fh | OSD3FIFOW | OSD3 FIFO Watermark | 0004h | R/W |
| 170h–171h | OSD3FIFOS | OSD3 FIFO Size | 00BFh | R/W |
| 172h–173h | OSD3FMT | OSD3 Format | 0000h | R/W |
| 178h–17Bh | OSD3BDPL | OSD3 Buffer Descriptor List Pointer-Lower Base Address | 00000000h | R/W, RO |
| 17Ch–17Fh | OSD3BDPU | OSD3 Buffer Description List Pointer-Upper Base Address | 00000000h | R/W |

## 18.2.1 GCAP—Global Capabilities Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 00h Attribute: RO
Default Value: 4401h Size: 16 bits

| Bit | Description |
|---|---|
| 15:12 | **Number of Output Stream Supported** — RO. Hardwired to 0100b indicating that the Chipset Intel HD Audio controller supports 4 output streams. |
| 11:8 | **Number of Input Stream Supported** — RO. Hardwired to 0100b indicating that the Chipset Intel High Definition Audio controller supports 4 input streams. |
| 7:3 | **Number of Bidirectional Stream Supported** — RO. Hardwired to 0 indicating that the Chipset Intel High Definition Audio controller supports 0 bidirectional stream. |
| 2 | **Reserved**. |
| 1 | **Number of Serial Data Out Signals** — RO. Hardwired to 0 indicating that the Chipset Intel High Definition Audio controller supports 1 serial data output signal. |
| 0 | **64-bit Address Supported** — RO. Hardwired to 1b indicating that the Chipset Intel High Definition Audio controller supports 64-bit addressing for BDL addresses, data buffer addressees, and command buffer addresses. |



### 18.2.2 VMIN—Minor Version Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 02h Attribute: RO

Default Value: 00hSize:8 bits

| Bit | Description |
|---|---|
| 7:0 | **Minor Version** — RO. Hardwired to 0 indicating that the Chipset supports minor revision number 00h of the Intel HD Audio specification. |

### 18.2.3 VMAJ—Major Version Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 03h Attribute: RO

Default Value: 01hSize:8 bits

| Bit | Description |
|---|---|
| 7:0 | **Major Version** — RO. Hardwired to 01h indicating that the Chipset supports major revision number 1 of the Intel HD Audio specification. |

### 18.2.4 OUTPAY—Output Payload Capability Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 04h Attribute: RO

Default Value: 003ChSize:16 bits

| Bit | Description |
|---|---|
| 15:7 | **Reserved**. |
| 6:0 | **Output Payload Capability** — RO. Hardwired to 3Ch indicating 60 word payload.<br>This field indicates the total output payload available on the link. This does not include bandwidth used for command and control. This measurement is in 16-bit word quantities per 48 MHz frame. The default link clock of 24.000 MHz (the data is double pumped) provides 1000 bits per frame, or 62.5 words in total. 40 bits are used for command and control, leaving 60 words available for data payload.<br>00h = 0 word<br>01h = 1 word payload.<br>.....<br>FFh = 256 word payload. |

### 18.2.5 INPAY—Input Payload Capability Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 06h Attribute: RO

Default Value: 001DhSize:16 bits

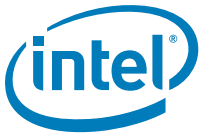
| Bit | Description |
|---|---|
| 15:7 | **Reserved**. |
| 6:0 | **Input Payload Capability** — RO. Hardwired to 1Dh indicating 29 word payload.<br>This field indicates the total output payload available on the link. This does not include bandwidth used for response. This measurement is in 16-bit word quantities per 48 MHz frame. The default link clock of 24.000 MHz provides 500 bits per frame, or 31.25 words in total. 36 bits are used for response, leaving 29 words available for data payload.<br>00h = 0 word<br>01h = 1 word payload.<br>.....<br>FFh = 256 word payload. |

### 18.2.6 GCTL—Global Control Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 08h Attribute: R/W

Default Value: 00000000hSize:32 bits

| Bit | Description |
|---|---|
| 31:9 | **Reserved**. |
| 8 | **Accept Unsolicited Response Enable** — R/W.<br>0 = Unsolicited responses from the codecs are not accepted.<br>1 = Unsolicited response from the codecs are accepted by the controller and placed into the Response Input Ring Buffer. |
| 7:2 | **Reserved**. |
| 1 | **Flush Control** — R/W.<br>0 = Flush **Not** in progress.<br>1 = Writing a 1 to this bit initiates a flush. When the flush completion is received by the controller, hardware sets the Flush Status bit and clears this Flush Control bit. Before a flush cycle is initiated, the DMA Position Buffer must be programmed with a valid memory address by software, but the DMA Position Buffer bit 0 needs not be set to enable the position reporting mechanism. Also, all streams must be stopped (the associated RUN bit must be 0).<br>When the flush is initiated, the controller will flush the pipelines to memory to ensure that the hardware is ready to transition to a D3 state. Setting this bit is not a critical step in the power state transition if the content of the FIFIOs is not critical. |



| Bit | Description |
|---|---|
| 0 | **Controller Reset #** — R/W.<br>0 = Writing a 0 to this bit causes the Intel High Definition Audio controller to be reset. All state machines, FIFOs, and non-resume well memory mapped configuration registers (not PCI configuration registers) in the controller will be reset. The Intel High Definition Audio link RESET# signal will be asserted, and all other link signals will be driven to their default values. After the hardware has completed sequencing into the reset state, it will report a 0 in this bit. Software must read a 0 from this bit to verify the controller is in reset.<br>1 = Writing a 1 to this bit causes the controller to exit its reset state and deassert the Intel High Definition Audio link RESET# signal. Software is responsible for setting/clearing this bit such that the minimum Intel High Definition Audio link RESET# signal assertion pulse width specification is met. When the controller hardware is ready to begin operation, it will report a 1 in this bit. Software must read a 1 from this bit before accessing any controller registers. This bit defaults to a 0 after Hardware reset, therefore, software needs to write a 1 to this bit to begin operation.<br><br>**NOTES:**<br>1. The CORB/RIRB RUN bits and all stream RUN bits must be verified cleared to 0 before writing a 0 to this bit in order to assure a clean re-start.<br>2. When setting or clearing this bit, software must ensure that minimum link timing requirements (minimum RESET# assertion time, etc.) are met.<br>3. When this bit is 0 indicating that the controller is in reset, writes to all Intel High Definition Audio memory mapped registers are ignored as if the device is not present. The only exception is this register itself. The Global Control register is write-able as a DWord, Word, or Byte even when CRST# (this bit) is 0 if the byte enable for the byte containing the CRST# bit (Byte Enable 0) is active. If Byte Enable 0 is not active, writes to the Global Control register will |

## 18.2.7 WAKEEN—Wake Enable Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 0Ch Attribute: R/W

Default Value: 0000hSize:16 bits

| Bit | Description |
|---|---|
| 15:3 | **Reserved**. |
| 2:0 | **SDIN Wake Enable Flags** — R/W. These bits control which SDI signal(s) may generate a wake event. A 1b in the bit mask indicates that the associated SDIN signal is enabled to generate a wake.<br>Bit 0 is used for SDI0<br>Bit 1 is used for SDI1<br>Bit 2 is used for SDI2<br><br>**NOTE:** These bits are in the resume well and only cleared on a power on reset. Software must not make assumptions about the reset state of these bits and must set them appropriately. |

### 18.2.8 STATESTS—State Change Status Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 0Eh Attribute: R/WC

Default Value: 0000hSize:16 bits

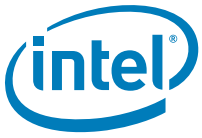
| Bit | Description |
|---|---|
| 15:3 | **Reserved**. |
| 2:0 | **SDIN State Change Status Flags** — R/WC. Flag bits that indicate which SDI signal(s) received a state change event. The bits are cleared by writing 1's to them.<br>Bit 0 = SDI0<br>Bit 1 = SDI1<br>Bit 2 = SDI2<br><br>**NOTE:** These bits are in the resume well and only cleared on a power on reset. Software must not make assumptions about the reset state of these bits and must set them appropriately. |

### 18.2.9 GSTS—Global Status Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 10h Attribute: R/WC

Default Value: 0000hSize:16 bits

| Bit | Description |
|---|---|
| 15:2 | **Reserved**. |
| 1 | **Flush Status** — R/WC.<br>0 = Flush **not** completed<br>1 = This bit is set to 1 by hardware to indicate that the flush cycle initiated when the Flush Control bit (HDBAR + 08h, bit 1) was set has completed.<br><br>**NOTE:** Software must write a 1 to clear this bit before the next time the Flush Control bit is set to clear the bit. |
| 0 | **Reserved**. |



## 18.2.10 OUTSTRMPAY—Output Stream Payload Capability (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 18h Attribute: RO

Default Value: 0030hSize:16 bits

| Bit | Description |
|---|---|
| 15:14 | **Output FIFO Padding Type (OPADTYPE)** — RO. This field indicates how the controller pads the samples in the controller's buffer (FIFO). Controllers may not pad at all or may pad to byte or memory container sizes.<br>0h = Controller pads all samples to bytes<br>1h = **Reserved**<br>2h = Controller pads to memory container size<br>3h = Controller does not pad and uses samples directly |
| 13:0 | **Output Stream Payload Capability (OUTSTRMPAY)** — RO. This field indicates maximum number of words per frame for any single output stream. This measurement is in 16 bit word quantities per 48 kHz frame. The maximum supported is 48 Words (96B); therefore, a value of 30h is reported in this register. The value does not specify the number of words actually transmitted in the frame, but is the size of the data in the controller buffer (FIFO) after the samples are padded as specified by OPADTYPE. Thus, to compute the supported streams, each sample is padded according to OPADTYPE and then multiplied by the number of channels and samples per frame. If this computed value is larger than OUTSTRMPAY, then that stream is not supported. The value specified is not affected by striping.<br>Software must ensure that a format that would cause more Words per frame than indicated is not programmed into the Output Stream Descriptor Register.<br>The value may be larger than the OUTPAY register value in some cases. |

## 18.2.11 INSTRMPAY—Input Stream Payload Capability (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 1Ah Attribute: RO

Default Value: 0018hSize:16 bits

| Bit | Description |
|---|---|
| 15:14 | **Input FIFO Padding Type (IPADTYPE)** — RO. This field indicates how the controller pads the samples in the controller's buffer (FIFO). Controllers may not pad at all or may pad to byte or memory container sizes.<br>0h = Controller pads all samples to bytes<br>1h = **Reserved**<br>2h = Controller pads to memory container size<br>3h = Controller does not pad and uses samples directly |

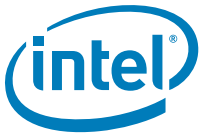
| Bit | Description |
|---|---|
| 13:0 | **Input Stream Payload Capability (INSTRMPAY)** — RO. This field indicates the maximum number of Words per frame for any single input stream. This measurement is in 16-bit Word quantities per<br>48-kHz frame. The maximum supported is 24 Words (48B); therefore, a value of 18h is reported in this register.<br>The value does not specify the number of words actually transmitted in the frame, but is the size of the data as it will be placed into the controller's buffer (FIFO). Thus, samples will be padded according to IPADTYPE before being stored into controller buffer. To compute the supported streams, each sample is padded according to IPADTYPE and then multiplied by the number of channels and samples per frame. If this computed value is larger than INSTRMPAY, then that stream is not supported. As the inbound stream tag is not stored with the samples it is not included in the word count.<br>The value may be larger than INPAY register value in some cases, although values less than INPAY may also be invalid due to overhead. Software must ensure that a format that would cause more Words per frame than indicated is not programmed into the Input Stream Descriptor Register. |

## 18.2.12 INTCTL—Interrupt Control Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 20h Attribute: R/W

Default Value: 00000000hSize:32 bits

| Bit | Description |
|---|---|
| 31 | **Global Interrupt Enable (GIE)** — R/W. Global bit to enable device interrupt generation.<br>0 = Disable.<br>1 = Enable. The Intel HD Audio function is enabled to generate an interrupt. This control is in addition to any bits in the bus specific address space, such as the Interrupt Enable bit in the PCI configuration space.<br><br>**NOTE:** This bit is not affected by the $D3_{HOT}$ to D0 transition. |
| 30 | **Controller Interrupt Enable (CIE)** — R/W. Enables the general interrupt for controller functions.<br>0 = Disable.<br>1 = Enable. The controller generates an interrupt when the corresponding status bit gets set due to a Response Interrupt, a Response Buffer Overrun, and State Change events.<br><br>**NOTE:** This bit is not affected by the $D3_{HOT}$ to D0 transition. |
| 29:8 | **Reserved** |



| Bit | Description |
|---|---|
| 7:0 | **Stream Interrupt Enable (SIE)** — R/W.<br>0 = Disable.<br>1 = Enable. When set to 1, the individual streams are enabled to generate an interrupt when the corresponding status bits get set.<br>A stream interrupt will be caused as a result of a buffer with IOC = 1in the BDL entry being completed, or as a result of a FIFO error (underrun or overrun) occurring. Control over the generation of each of these sources is in the associated Stream Descriptor.<br>The streams are numbered and the SIE bits assigned sequentially, based on their order in the register set.<br>Bit 0 = input stream 1<br>Bit 1 = input stream 2<br>Bit 2 = input stream 3<br>Bit 3 = input stream 4<br>Bit 4 = output stream 1<br>Bit 5 = output stream 2<br>Bit 6 = output stream 3<br>Bit 7 = output stream 4 |

## 18.2.13 INTSTS—Interrupt Status Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 24h Attribute: RO

Default Value: 00000000hSize:32 bits

| Bit | Description |
|---|---|
| 31 | **Global Interrupt Status (GIS)** — RO. This bit is an OR of all the interrupt status bits in this register.<br>**NOTE:** This bit is not affected by the $D3_{HOT}$ to D0 transition. |
| 30 | **Controller Interrupt Status (CIS)** — RO. Status of general controller interrupt.<br>0 = An interrupt condition did **Not** occur as described below.<br>1 = An interrupt condition occurred due to a Response Interrupt, a Response Buffer Overrun Interrupt, or a SDIN State Change event. The exact cause can be determined by interrogating other registers. This bit is an OR of all of the stated interrupt status bits for this register.<br><br>**NOTES:**<br>1. This bit is set regardless of the state of the corresponding interrupt enable bit, but a hardware interrupt will not be generated unless the corresponding enable bit is set.<br>2. This bit is not affected by the $D3_{HOT}$ to D0 transition. |
| 29:8 | **Reserved** |

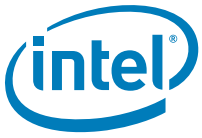
| Bit | Description |
|---|---|
| 7:0 | **Stream Interrupt Status (SIS)** — RO.<br>0 = An interrupt condition did **Not** occur on the corresponding stream.<br>1 = An interrupt condition occurred on the corresponding stream. This bit is an OR of all of the stream's interrupt status bits.<br><br>**NOTE:** These bits are set regardless of the state of the corresponding interrupt enable bits.<br>The streams are numbered and the SIE bits assigned sequentially, based on their order in the register set.<br>Bit 0 = input stream 1<br>Bit 1 = input stream 2<br>Bit 2 = input stream 3<br>Bit 3 = input stream 4<br>Bit 4 = output stream 1<br>Bit 5 = output stream 2<br>Bit 6 = output stream 3<br>Bit 7 = output stream 4 |

## 18.2.14 WALCLK—Wall Clock Counter Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 30h Attribute: RO

Default Value: 00000000hSize:32 bits

| Bit | Description |
|---|---|
| 31:0 | **Wall Clock Counter** — RO. This 32-bit counter field is incremented on each link BCLK period and rolls over from FFFF FFFFh to 0000 0000h. This counter will roll over to 0 with a period of approximately 179 seconds.<br>This counter is enabled while the BCLK bit is set to 1. Software uses this counter to synchronize between multiple controllers. Will be reset on controller reset. |



## 18.2.15 SSYNC—Stream Synchronization Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 34h Attribute: R/W

Default Value: 00000000hSize:32 bits

| Bit | Description |
|---|---|
| 31:8 | **Reserved** |
| 7:0 | **Stream Synchronization (SSYNC)** — R/W.<br>0 = Data is **Not** blocked from being sent on or received from the link<br>1 = The set bits block data from being sent on or received from the link. Each bit controls the associated stream descriptor (i.e., bit 0 corresponds to the first stream descriptor, etc.)<br>To synchronously start a set of DMA engines, these bits are first set to 1. The RUN bits for the associated stream descriptors are then set to 1 to start the DMA engines. When all streams are ready (FIFORDY =1), the associated SSYNC bits can all be set to 0 at the same time, and transmission or reception of bits to or from the link will begin together at the start of the next full link frame.<br>To synchronously stop the streams, first these bits are set, and then the individual RUN bits in the stream descriptor are cleared by software.<br>If synchronization is not desired, these bits may be left as 0, and the stream will simply begin running normally when the stream's RUN bit is set.<br>The streams are numbered and the SIE bits assigned sequentially, based on their order in the register set.<br>Bit 0 = input stream 1<br>Bit 1 = input stream 2<br>Bit 2 = input stream 3<br>Bit 3 = input stream 4<br>Bit 4 = output stream 1<br>Bit 5 = output stream 2<br>Bit 6 = output stream 3<br>Bit 7 = output stream 4 |

## 18.2.16 CORBLBASE—CORB Lower Base Address Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 40h Attribute: R/W, RO

Default Value: 00000000hSize:32 bits

| Bit | Description |
|---|---|
| 31:7 | **CORB Lower Base Address** — R/W. This field is the lower address of the Command Output Ring Buffer, allowing the CORB base address to be assigned on any 128-B boundary. This register field must not be written when the DMA engine is running or the DMA transfer may be corrupted. |
| 6:0 | **CORB Lower Base Unimplemented Bits** — RO. Hardwired to 0. This requires the CORB to be allocated with 128B granularity to allow for cache line fetch optimizations. |

## 18.2.17 CORBUBASE—CORB Upper Base Address Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 44h Attribute: R/W

Default Value: 00000000hSize:32 bits

| Bit | Description |
|---|---|
| 31:0 | **CORB Upper Base Address** — R/W. This field is the upper 32 bits of the address of the Command Output Ring buffer. This register field must not be written when the DMA engine is running or the DMA transfer may be corrupted. |

## 18.2.18 CORBWP—CORB Write Pointer Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 48h Attribute: R/W

Default Value: 0000hSize:16 bits

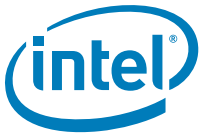
| Bit | Description |
|---|---|
| 15:8 | **Reserved**. |
| 7:0 | **CORB Write Pointer** — R/W. Software writes the last valid CORB entry offset into this field in DWord granularity. The DMA engine fetches commands from the CORB until the Read pointer matches the Write pointer. Supports 256 CORB entries (256x4B = 1KB). This register field may be written when the DMA engine is running. |

## 18.2.19 CORBRP—CORB Read Pointer Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 4Ah Attribute: R/W

Default Value: 0000hSize:16 bits

| Bit | Description |
|---|---|
| 15 | **CORB Read Pointer Reset** — R/W. Software writes a 1 to this bit to reset the CORB Read Pointer to 0 and clear any residual prefetched commands in the CORB hardware buffer within the Intel HD Audio controller. The hardware will physically update this bit to 1 when the CORB Pointer reset is complete. Software must read a 1 to verify that the reset completed correctly. Software must clear this bit back to 0 and read back the 0 to verify that the clear completed correctly. The CORB DMA engine must be stopped prior to resetting the Read Pointer or else DMA transfer may be corrupted. |
| 14:8 | **Reserved**. |
| 7:0 | **CORB Read Pointer (CORBRP)** — RO. Software reads this field to determine how many commands it can write to the CORB without over-running. The value read indicates the CORB Read Pointer offset in DWord granularity. The offset entry read from this field has been successfully fetched by the DMA controller and may be over-written by software. Supports 256 CORB entries (256 x 4B=1KB). This field may be read while the DMA engine is running. |



## 18.2.20 CORBCTL—CORB Control Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 4Ch Attribute: R/W

Default Value: 00hSize:8 bits

| Bit | Description |
|---|---|
| 7:2 | **Reserved**. |
| 1 | **Enable CORB DMA Engine** — R/W. After software writes a 0 to this bit, the hardware may not stop immediately. The hardware will physically update the bit to 0 when the DMA engine is truly stopped. Software must read a 0 from this bit to verify that the DMA engine is truly stopped.<br>0 = DMA stop<br>1 = DMA run |
| 0 | **CORB Memory Error Interrupt Enable** — R/W.<br>0 = Disable.<br>1 = Enable. The controller will generate an interrupt if the CMEI status bit (HDBAR + 4Dh: bit 0) is set. |

## 18.2.21 CORBST—CORB Status Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 4Dh Attribute: R/WC

Default Value: 00hSize:8 bits

| Bit | Description |
|---|---|
| 7:1 | **Reserved**. |
| 0 | **CORB Memory Error Indication (CMEI)** — R/WC.<br>0 = Error **Not** detected.<br>1 = The controller has detected an error in the path way between the controller and memory. This may be an ECC bit error or any other type of detectable data error which renders the command data fetched invalid.<br><br>**NOTE:** Software can clear this bit by writing a 1 to it. However, this type of error leaves the audio subsystem in an un-viable state and typically requires a controller reset by writing a 0 to the Controller Reset # bit (HDBAR + 08h: bit 0). |

## 18.2.22 CORBSIZE—CORB Size Register Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 4Eh Attribute: RO

Default Value: 42hSize:8 bits

| Bit | Description |
|---|---|
| 7:4 | **CORB Size Capability** — RO. Hardwired to 0100b indicating that the Chipset only supports a CORB size of 256 CORB entries (1024B). |
| 3:2 | **Reserved**. |
| 1:0 | **CORB Size** — RO. Hardwired to 10b which sets the CORB size to 256 entries (1024B). |

### 18.2.23 RIRBLBASE—RIRB Lower Base Address Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 50h Attribute: R/W, RO

Default Value: 00000000hSize:32 bits

| Bit | Description |
|---|---|
| 31:7 | **RIRB Lower Base Address** — R/W. This field is the lower address of the Response Input Ring Buffer, allowing the RIRB base address to be assigned on any 128-B boundary. This register field must not be written when the DMA engine is running or the DMA transfer may be corrupted. |
| 6:0 | RIRB Lower Base Unimplemented Bits — RO. Hardwired to 0. This required the RIRB to be allocated with 128-B granularity to allow for cache line fetch optimizations. |

### 18.2.24 RIRBUBASE—RIRB Upper Base Address Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 54h Attribute: R/W

Default Value: 00000000hSize:32 bits

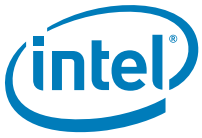
| Bit | Description |
|---|---|
| 31:0 | **RIRB Upper Base Address** — R/W. This field is the upper 32 bits of the address of the Response Input Ring Buffer. This register field must not be written when the DMA engine is running or the DMA transfer may be corrupted. |

### 18.2.25 RIRBWP—RIRB Write Pointer Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 58h Attribute: R/W, RO

Default Value: 0000hSize:16 bits

| Bit | Description |
|---|---|
| 15 | **RIRB Write Pointer Reset** — R/W. Software writes a 1 to this bit to reset the RIRB Write Pointer to 0. The RIRB DMA engine must be stopped prior to resetting the Write Pointer or else DMA transfer may be corrupted.<br>This bit is always read as 0. |
| 14:8 | **Reserved**. |
| 7:0 | **RIRB Write Pointer (RIRBWP)** — RO. This field is the indicates the last valid RIRB entry written by the DMA controller. Software reads this field to determine how many responses it can read from the RIRB. The value read indicates the RIRB Write Pointer offset in 2 DWord RIRB entry units (since each RIRB entry is 2 DWords long). Supports up to 256 RIRB entries (256 x 8 B = 2 KB). This register field may be written when the DMA engine is running. |



## 18.2.26 RINTCNT—Response Interrupt Count Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 5Ah Attribute: R/W

Default Value: 0000hSize:16 bits

| Bit | Description |
|---|---|
| 15:8 | **Reserved**. |
| 7:0 | **N Response Interrupt Count** — R/W.<br>0000 0001b = 1 response sent to RIRB<br>.....<br>1111 1111b = 255 responses sent to RIRB<br>0000 0000b = 256 responses sent to RIRB<br>The DMA engine should be stopped when changing this field or else an interrupt may be lost.<br>Note that each response occupies 2 DWords in the RIRB.<br>This is compared to the total number of responses that have been returned, as opposed to the number of frames in which there were responses. If more than one codec responds in one frame, then the count is increased by the number of responses received in the frame. |

## 18.2.27 RIRBCTL—RIRB Control Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 5Ch Attribute: R/W

Default Value: 00hSize:8 bits

| Bit | Description |
|---|---|
| 7:3 | **Reserved**. |
| 2 | **Response Overrun Interrupt Control** — R/W.<br>0 = Hardware will **Not** generated an interrupt as described below.<br>1 = The hardware will generate an interrupt when the Response Overrun Interrupt Status bit (HDBAR + 5Dh: bit 2) is set. |
| 1 | **Enable RIRB DMA Engine** — R/W. After software writes a 0 to this bit, the hardware may not stop immediately. The hardware will physically update the bit to 0 when the DMA engine is truly stopped. Software must read a 0 from this bit to verify that the DMA engine is truly stopped.<br>0 = DMA stop<br>1 = DMA run |
| 0 | **Response Interrupt Control** — R/W.<br>0 = Disable Interrupt<br>1 = Generate an interrupt after N number of responses are sent to the RIRB buffer OR when an empty Response slot is encountered on all SDI[x] inputs (whichever occurs first). The N counter is reset when the interrupt is generated. |

### 18.2.28 RIRBSTS—RIRB Status Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 5Dh Attribute: R/WC

Default Value: 00hSize:8 bits

| Bit | Description |
|---|---|
| 7:3 | **Reserved**. |
| 2 | **Response Overrun Interrupt Status** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Software sets this bit to 1 when the RIRB DMA engine is not able to write the incoming responses to memory before additional incoming responses overrun the internal FIFO. When the overrun occurs, the hardware will drop the responses that overrun the buffer. An interrupt may be generated if the Response Overrun Interrupt Control bit is set. Note that this status bit is set even if an interrupt is not enabled for this event. |
| 1 | **Reserved**. |
| 0 | **Response Interrupt** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Hardware sets this bit to 1 when an interrupt has been generated after N number of Responses are sent to the RIRB buffer OR when an empty Response slot is encountered on all SDI[x] inputs (whichever occurs first). Note that this status bit is set even if an interrupt is not enabled for this event. |

### 18.2.29 RIRBSIZE—RIRB Size Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 5Eh Attribute: RO

Default Value: 42hSize:8 bits

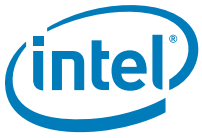
| Bit | Description |
|---|---|
| 7:4 | **RIRB Size Capability** — RO. Hardwired to 0100b indicating that the Chipset only supports a RIRB size of 256 RIRB entries (2048B) |
| 3:2 | **Reserved**. |
| 1:0 | **RIRB Size — RO**. Hardwired to 10b which sets the CORB size to 256 entries (2048B) |

### 18.2.30 IC—Immediate Command Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 60h Attribute: R/W

Default Value: 00000000hSize:32 bits

| Bit | Description |
|---|---|
| 31:0 | **Immediate Command Write — R/W**. The command to be sent to the codec via the Immediate Command mechanism is written to this register. The command stored in this register is sent out over the link during the next available frame after a 1 is written to the ICB bit (HDBAR + 68h: bit 0) |



## 18.2.31 IR—Immediate Response Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 64h Attribute: RO

Default Value: 00000000hSize:32 bits

| Bit | Description |
|---|---|
| 31:0 | **Immediate Response Read (IRR)** — RO. This register contains the response received from a codec resulting from a command sent via the Immediate Command mechanism.<br>If multiple codecs responded in the same time, there is no assurance as to which response will be latched. Therefore, broadcast-type commands must not be issued via the Immediate Command mechanism. |

## 18.2.32 IRS—Immediate Command Status Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 68h Attribute: R/W, R/WC

Default Value: 0000hSize:16 bits

| Bit | Description |
|---|---|
| 15:2 | **Reserved**. |
| 1 | **Immediate Result Valid (IRV)** — R/WC.<br>0 = Software must clear this bit by writing a 1 to it before issuing a new command so that the software may determine when a new response has arrived.<br>1 = Set to 1 by hardware when a new response is latched into the Immediate Response register (HDBAR + 64). This is a status flag indicating that software may read the response from the Immediate Response register. |
| 0 | **Immediate Command Busy (ICB)** — R/W. When this bit is read as 0, it indicates that a new command may be issued using the Immediate Command mechanism. When this bit transitions from 0-to-1 (via software writing a 1), the controller issues the command currently stored in the Immediate Command register to the codec over the link. When the corresponding response is latched into the Immediate Response register, the controller hardware sets the IRV flag and clears the ICB bit back to 0.<br><br>**NOTE:** An Immediate Command must not be issued while the CORB/RIRB mechanism is operating, otherwise the responses conflict. This must be enforced by software. |

## 18.2.33 DPLBASE—DMA Position Lower Base Address Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 70h Attribute: R/W, RO

Default Value: 00000000hSize:32 bits

| Bit | Description |
|---|---|
| 31:7 | **DMA Position Lower Base Address** — R/W. Lower 32 bits of the DMA Position Buffer Base Address. This register field must not be written when any DMA engine is running or the DMA transfer may be corrupted. This same address is used by the Flush Control and must be programmed with a valid value before the Flush Control bit (HDBAR+08h:bit 1) is set. |
| 6:1 | DMA Position Lower Base Unimplemented bits — RO. Hardwired to 0 to force the 128-byte buffer alignment for cache line write optimizations. |
| 0 | **DMA Position Buffer Enable** — R/W.<br>0 = Disable.<br>1 = Enable. The controller will write the DMA positions of each of the DMA engines to the buffer in the main memory periodically (typically once per frame). Software can use this value to determine what data in memory is valid data. |

## 18.2.34 DPUBASE—DMA Position Upper Base Address Register (Intel HD Audio Controller—D27:F0)

Memory Address:HDBAR + 74h Attribute: R/W

Default Value:00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **DMA Position Upper Base Address** — R/W. Upper 32 bits of the DMA Position Buffer Base Address. This register field must not be written when any DMA engine is running or the DMA transfer may be corrupted. |

## 18.2.35 SDCTL—Stream Descriptor Control Register (Intel HD Audio Controller—D27:F0)

Memory Address:Input Stream[0]: HDBAR + 80hAttribute:R/W, RO
Input Stream[1]: HDBAR + A0h
Input Stream[2]: HDBAR + C0h
Input Stream[3]: HDBAR + E0h
Output Stream[0]: HDBAR + 100h
Output Stream[1]: HDBAR + 120h
Output Stream[2]: HDBAR + 140h
Output Stream[3]: HDBAR + 160h

Default Value: 040000h Size: 24 bits

| Bit | Description |
|---|---|
| 23:20 | **Stream Number** — R/W. This value reflects the Tag associated with the data being transferred on the link.<br>When data controlled by this descriptor is sent out over the link, it will have its stream number encoded on the SYNC signal.<br>When an input stream is detected on any of the SDI signals that match this value, the data samples are loaded into FIFO associated with this descriptor.<br>Note that while a single SDI input may contain data from more than one stream number, two different SDI inputs may not be configured with the same stream number.<br>0000 = **Reserved**<br>0001 = Stream 1<br>.....<br>1110 = Stream 14<br>1111 = Stream 15 |
| 19 | **Bidirectional Direction Control** — RO. This bit is only meaningful for bidirectional streams; therefore, this bit is hardwired to 0. |
| 18 | **Traffic Priority** — RO. Hardwired to 1 indicating that all streams will use VC1 if it is enabled through the PCI Express* registers. |
| 17:16 | **Stripe Control** — RO. This bit is only meaningful for input streams; therefore, this bit is hardwired to 0. |
| 15:5 | **Reserved** |
| 4 | **Descriptor Error Interrupt Enable** — R/W.<br>0 = Disable<br>1 = An interrupt is generated when the Descriptor Error Status bit is set. |
| 3 | **FIFO Error Interrupt Enable** — R/W.<br>0 = Disable.<br>1 = Enable. This bit controls whether the occurrence of a FIFO error (overrun for input or underrun for output) will cause an interrupt or not. If this bit is not set, bit 3 in the Status register will be set, but the interrupt will not occur. Either way, the samples will be dropped. |
| 2 | **Interrupt on Completion Enable** — R/W.<br>0 = Disable.<br>1 = Enable. This bit controls whether or not an interrupt occurs when a buffer completes with the IOC bit set in its descriptor. If this bit is not set, bit 2 in the Status register will be set, but the interrupt will not occur. |
| 1 | **Stream Run (RUN)** — R/W.<br>0 = Disable. The DMA engine associated with this input stream will be disabled. The hardware will report a 0 in this bit when the DMA engine is actually stopped. Software must read a 0 from this bit before modifying related control registers or restarting the DMA engine.<br>1 = Enable. The DMA engine associated with this input stream will be enabled to transfer data from the FIFO to the main memory. The SSYNC bit must also be cleared in order for the DMA engine to run. For output streams, the cadence generator is reset whenever the RUN bit is set. |

| Bit | Description |
|---|---|
| 0 | **Stream Reset (SRST)** — R/W.<br>0 = Writing a 0 causes the corresponding stream to exit reset. When the stream hardware is ready to begin operation, it will report a 0 in this bit. Software must read a 0 from this bit before accessing any of the stream registers.<br>1 = Writing a 1 causes the corresponding stream to be reset. The Stream Descriptor registers (except the SRST bit itself) and FIFO's for the corresponding stream are reset. After the stream hardware has completed sequencing into the reset state, it will report a 1 in this bit. Software must read a 1 from this bit to verify that the stream is in reset. The RUN bit must be cleared before SRST is asserted. |

## 18.2.36 SDSTS—Stream Descriptor Status Register (Intel HD Audio Controller—D27:F0)

Memory Address: Input Stream[0]: HDBAR + 83h Attribute: R/WC, RO
Input Stream[1]: HDBAR + A3h
Input Stream[2]: HDBAR + C3h
Input Stream[3]: HDBAR + E3h
Output Stream[0]: HDBAR + 103h
Output Stream[1]: HDBAR + 123h
Output Stream[2]: HDBAR + 143h
Output Stream[3]: HDBAR + 163h

Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:6 | **Reserved**. |
| 5 | **FIFO Ready (FIFORDY)** — RO.<br>For output streams, the controller hardware will set this bit to 1 while the output DMA FIFO contains enough data to maintain the stream on the link. This bit defaults to 0 on reset because the FIFO is cleared on a reset.<br>For input streams, the controller hardware will set this bit to 1 when a valid descriptor is loaded and the engine is ready for the RUN bit to be set. |
| 4 | **Descriptor Error** — R/WC.<br>0 = No error detected.<br>1 = A serious error occurred during the fetch of a descriptor. This could be a result of a Master Abort, a parity or ECC error on the bus, or any other error which renders the current Buffer Descriptor or Buffer Descriptor list useless. This error is treated as a fatal stream error, as the stream cannot continue running. The RUN bit will be cleared and the stream will stop.<br><br>**NOTE:** Software may attempt to restart the stream engine after addressing the cause of the error and writing a 1 to this bit to clear it. |

| | |
|---|---|
| 3 | **FIFO Error** — R/WC. The bit is cleared by writing a 1 to it.<br>0 = No error detected.<br>1 = FIFO error occurred. This bit is set even if an interrupt is not enabled.<br>For an input stream, this indicates a FIFO overrun occurring while the RUN bit is set. When this happens, the FIFO pointers do not increment and the incoming data is not written into the FIFO, thereby being lost.<br>For an output stream, this indicates a FIFO underrun when there are still buffers to send. The hardware should not transmit anything on the link for the associated stream if there is not valid data to send. |
| 2 | **Buffer Completion Interrupt Status** — R/WC.<br>0 = Last sample of a buffer has **Not** been processed as described below.<br>1 = Set to 1 by the hardware after the last sample of a buffer has been processed, AND if the Interrupt on Completion bit is set in the command byte of the buffer descriptor. It remains active until software clears it by writing a 1 to it. |
| 1:0 | **Reserved**. |

## 18.2.37 SDLPIB—Stream Descriptor Link Position in Buffer Register (Intel HD Audio Controller—D27:F0)

Memory Address: Input Stream[0]: HDBAR + 84h — Attribute: RO
Input Stream[1]: HDBAR + A4h
Input Stream[2]: HDBAR + C4h
Input Stream[3]: HDBAR + E4h
Output Stream[0]: HDBAR + 104h
Output Stream[1]: HDBAR + 124h
Output Stream[2]: HDBAR + 144h
Output Stream[3]: HDBAR + 164h

Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Link Position in Buffer** — RO. Indicates the number of bytes that have been received off the link. This register will count from 0 to the value in the Cyclic Buffer Length register and then wrap to 0. |

## 18.2.38 SDCBL—Stream Descriptor Cyclic Buffer Length Register (Intel HD Audio Controller—D27:F0)

Memory Address: Input Stream[0]: HDBAR + 88h — Attribute: R/W
Input Stream[1]: HDBAR + A8h
Input Stream[2]: HDBAR + C8h
Input Stream[3]: HDBAR + E8h
Output Stream[0]: HDBAR + 108h
Output Stream[1]: HDBAR + 128h
Output Stream[2]: HDBAR + 148h
Output Stream[3]: HDBAR + 168h

Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Cyclic Buffer Length** — R/W. Indicates the number of bytes in the complete cyclic buffer. This register represents an integer number of samples. Link Position in Buffer will be reset when it reaches this value.<br>Software may only write to this register after Global Reset, Controller Reset, or Stream Reset has occurred. This value should be only modified when the RUN bit is 0. Once the RUN bit has been set to enable the engine, software must not write to this register until after the next reset is asserted, or transfer may be corrupted. |

## 18.2.39 SDLVI—Stream Descriptor Last Valid Index Register (Intel HD Audio Controller—D27:F0)

Memory Address: Input Stream[0]: HDBAR + 8Ch Attribute: R/W
Input Stream[1]: HDBAR + ACh
Input Stream[2]: HDBAR + CCh
Input Stream[3]: HDBAR + ECh
Output Stream[0]: HDBAR + 10Ch
Output Stream[1]: HDBAR + 12Ch
Output Stream[2]: HDBAR + 14Ch
Output Stream[3]: HDBAR + 16Ch

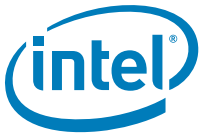
Default Value: 0000hSize:16 bits

| Bit | Description |
|---|---|
| 15:8 | **Reserved**. |
| 7:0 | **Last Valid Index** — R/W. The value written to this register indicates the index for the last valid Buffer Descriptor in BDL. After the controller has processed this descriptor, it will wrap back to the first descriptor in the list and continue processing.<br>This field must be at least 1 (i.e., there must be at least 2 valid entries in the buffer descriptor list before DMA operations can begin).<br>This value should only be modified when the RUN bit is 0. |

## 18.2.40 SDFIFOW—Stream Descriptor FIFO Watermark Register (Intel HD Audio Controller—D27:F0)

Memory Address: Input Stream[0]: HDBAR + 8Eh Attribute: R/W
Input Stream[1]: HDBAR + AEh
Input Stream[2]: HDBAR + CEh
Input Stream[3]: HDBAR + EEh
Output Stream[0]: HDBAR + 10Eh
Output Stream[1]: HDBAR + 12Eh
Output Stream[2]: HDBAR + 14Eh
Output Stream[3]: HDBAR + 16Eh

Default Value: 0004hSize:16 bits



| Bit | Description |
|---|---|
| 15:3 | **Reserved**. |
| 2:0 | **FIFO Watermark (FIFOW)** — R/W. This field indicates the minimum number of bytes accumulated/free in the FIFO before the controller will start a fetch/eviction of data.<br>010 = 8B<br>011 = 16B<br>100 = 32B (Default)<br>Others = Unsupported<br><br>**NOTES:**<br>1. When the bit field is programmed to an unsupported size, the hardware sets itself to the default value.<br>2. Software must read the bit field to test if the value is supported after setting the bit field. |

## 18.2.41 SDFIFOS—Stream Descriptor FIFO Size Register (Intel HD Audio Controller—D27:F0)

Memory Address: Input Stream[0]: HDBAR + 90h Attribute: Input: RO
Input Stream[1]: HDBAR + B0h Output: R/W
Input Stream[2]: HDBAR + D0h
Input Stream[3]: HDBAR + F0h
Output Stream[0]: HDBAR + 110h
Output Stream[1]: HDBAR + 130h
Output Stream[2]: HDBAR + 150h
Output Stream[3]: HDBAR + 170h

Default Value: Input Stream: 0077h Size: 16 bits
Output Stream: 00BFh

| Bit | Description |
|---|---|
| 15:8 | **Reserved**. |
| 7:0 | **FIFO Size** — RO (Input stream), R/W (Output stream). This field indicates the maximum number of bytes that could be fetched by the controller at one time. This is the maximum number of bytes that may have been DMA'd into memory but not yet transmitted on the link, and is also the maximum possible value that the PICB count will increase by at one time.<br>The value in this field is different for input and output streams. It is also dependent on the Bits per Samples setting for the corresponding stream. Following are the values read/written from/to this register for input and output streams, and for non-padded and padded bit formats:<br>*Output Stream R/W value*:<br><br>**NOTES:**<br>1. All other values not listed are not supported.<br>2. When the output stream is programmed to an unsupported size, the hardware sets itself to the default value (BFh).<br>3. Software must read the bit field to test if the value is supported after setting the bit field.<br>*Input Stream RO value*:<br><br>**NOTE:** The default value is different for input and output streams, and reflects the default state of the BITS fields (in Stream Descriptor Format registers) for the corresponding stream. |

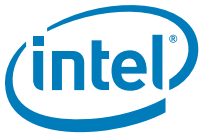
| Value | Output Streams |
|---|---|
| 0Fh = 16B | 8, 16, 20, 24, or 32 bit Output Streams |
| 1Fh = 32B | 8, 16, 20, 24, or 32 bit Output Streams |
| 3Fh = 64B | 8, 16, 20, 24, or 32 bit Output Streams |
| 7Fh = 128B | 8, 16, 20, 24, or 32 bit Output Streams |
| BFh = 192B | 8, 16, or 32 bit Output Streams |
| FFh = 256B | 20, 24 bit Output Streams |

| Value | Input Streams |
|---|---|
| 77h = 120B | 8, 16, 32 bit Input Streams |
| 9Fh = 160B | 20, 24 bit Input Streams |

## 18.2.42 SDFMT—Stream Descriptor Format Register (Intel HD Audio Controller—D27:F0)

Memory Address: Input Stream[0]: HDBAR + 92h Attribute: R/W
Input Stream[1]: HDBAR + B2h
Input Stream[2]: HDBAR + D2h
Input Stream[3]: HDBAR + F2h
Output Stream[0]: HDBAR + 112h
Output Stream[1]: HDBAR + 132h
Output Stream[2]: HDBAR + 152h
Output Stream[3]: HDBAR + 172h

Default Value: 0000h Size: 16 bits



| Bit | Description |
|---|---|
| 15 | **Reserved**. |
| 14 | **Sample Base Rate** — R/W<br>0 = 48 kHz<br>1 = 44.1 kHz |
| 13:11 | **Sample Base Rate Multiple** — R/W<br>000 = 48 kHz, 44.1 kHz or less<br>001 = x2 (96 kHz, 88.2 kHz, 32 kHz)<br>010 = x3 (144 kHz)<br>011 = x4 (192 kHz, 176.4 kHz)<br>Others = **Reserved**. |
| 10:8 | **Sample Base Rate Divisor** — R/W.<br>000 = Divide by 1(48 kHz, 44.1 kHz)<br>001 = Divide by 2 (24 kHz, 22.05 kHz)<br>010 = Divide by 3 (16 kHz, 32 kHz)<br>011 = Divide by 4 (11.025 kHz)<br>100 = Divide by 5 (9.6 kHz)<br>101 = Divide by 6 (8 kHz)<br>110 = Divide by 7<br>111 = Divide by 8 (6 kHz) |
| 7 | **Reserved**. |
| 6:4 | **Bits per Sample (BITS)** — R/W.<br>000 = 8 bits. The data will be packed in memory in 8-bit containers on 16-bit boundaries<br>001 = 16 bits. The data will be packed in memory in 16-bit quantities on 16-bit boundaries<br>010 = 20 bits. The data will be packed in memory in 32-bit quantities on 32-bit boundaries<br>011 = 24 bits. The data will be packed in memory in 32-bit quantities on 32-bit boundaries<br>100 = 32 bits. The data will be packed in memory in 32-bit quantities on 32-bit boundaries<br>Others = **Reserved**. |
| 3:0 | Number of Channels (CHAN) — R/W. Indicates number of channels in each frame of the stream.<br>0000 =1<br>0001 =2<br>……..<br>1111 =16 |

## 18.2.43 SDBDPL—Stream Descriptor Buffer Descriptor List Pointer Lower Base Address Register (Intel HD Audio Controller—D27:F0)

Memory Address: Input Stream[0]: HDBAR + 98h Attribute: R/W,RO
Input Stream[1]: HDBAR + B8h
Input Stream[2]: HDBAR + D8h
Input Stream[3]: HDBAR + F8h
Output Stream[0]: HDBAR + 118h
Output Stream[1]: HDBAR + 138h
Output Stream[2]: HDBAR + 158h
Output Stream[3]: HDBAR + 178h

Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:7 | **Buffer Descriptor List Pointer Lower Base Address** — R/W. This field is the lower address of the Buffer Descriptor List. This value should only be modified when the RUN bit is 0, or DMA transfer may be corrupted. |
| 6:0 | Hardwired to 0 forcing alignment on 128-B boundaries. |

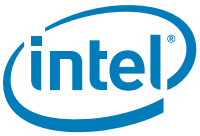
## 18.2.44 SDBDPU—Stream Descriptor Buffer Descriptor List Pointer Upper Base Address Register (Intel HD Audio Controller—D27:F0)

Memory Address: Input Stream[0]: HDBAR + 9Ch Attribute: R/W
Input Stream[1]: HDBAR + BCh
Input Stream[2]: HDBAR + DCh
Input Stream[3]: HDBAR + FCh
Output Stream[0]: HDBAR + 11Ch
Output Stream[1]: HDBAR + 13Ch
Output Stream[2]: HDBAR + 15Ch
Output Stream[3]: HDBAR + 17Ch

Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Buffer Descriptor List Pointer Upper Base Address** — R/W. This field is the upper 32-bit address of the Buffer Descriptor List. This value should only be modified when the RUN bit is 0, or DMA transfer may be corrupted. |

§



# 19 PCI Express* Configuration Registers

## 19.1 PCI Express* Configuration Registers (PCI Express—D28:F0/F1/F2/F3)

***Note:*** Register address locations that are not shown in Table 19-148 should be treated as **Reserved**.

**Table 19-148.PCI Express* Configuration Registers Address Map (PCI Express—D28:F0/F1/F2/F3) (Sheet 1 of 3)**

| Offset | Mnemonic | Register Name | Function 0–5 Default | Type |
|---|---|---|---|---|
| 00h–01h | VID | Vendor Identification | 8086h | RO |
| 02h–03h | DID | Device Identification | See register description. | RO |
| 04h–05h | PCICMD | PCI Command | 0000h | R/W, RO |
| 06h–07h | PCISTS | PCI Status | 0010h | R/WC, RO |
| 08h | RID | Revision Identification | See register description. | RO |
| 09h | PI | Programming Interface | 00h | RO |
| 0Ah | SCC | Sub Class Code | 04h | RO |
| 0Bh | BCC | Base Class Code | 06h | RO |
| 0Ch | CLS | Cache Line Size | 00h | R/W |
| 0Dh | PLT | Primary Latency Timer | 00h | RO |
| 0Eh | HEADTYP | Header Type | 81h | RO |
| 18h–1Ah | BNUM | Bus Number | 000000h | R/W |
| 1Ch–1Dh | IOBL | I/O Base and Limit | 0000h | R/W, RO |
| 1Eh–1Fh | SSTS | Secondary Status | 0000h | R/WC |
| 20h–23h | MBL | Memory Base and Limit | 00000000h | R/W |
| 24h–27h | PMBL | Prefetchable Memory Base and Limit | 00010001h | R/W, RO |
| 28h–2Bh | PMBU32 | Prefetchable Memory Base Upper 32 Bits | 00000000h | R/W |
| 2Ch–2Fh | PMLU32 | Prefetchable Memory Limit Upper 32 Bits | 00000000h | R/W |
| 34h | CAPP | Capabilities List Pointer | 40h | RO |
| 3Ch–3Dh | INTR | Interrupt Information | See bit description | R/W, RO |
| 3Eh–3Fh | BCTRL | Bridge Control | 0000h | R/W |
| 40h–41h | CLIST | Capabilities List | 8010 | RO |
| 42h–43h | XCAP | PCI Express* Capabilities | 0041 | R/WO, RO |
| 44h–47h | DCAP | Device Capabilities | 00000FE0h | RO |
| 48h–49h | DCTL | Device Control | 0000h | R/W, RO |
| 4Ah–4Bh | DSTS | Device Status | 0010h | R/WC, RO |

**Table 19-148.PCI Express* Configuration Registers Address Map (PCI Express—D28:F0/F1/F2/F3) (Sheet 2 of 3)**

| Offset | Mnemonic | Register Name | Function 0–5 Default | Type |
|---|---|---|---|---|
| 4Ch–4Fh | LCAP | Link Capabilities | See bit description | R/W, RO, R/WO |
| 50h–51h | LCTL | Link Control | 0000h | R/W, WO, RO |
| 52h–53h | LSTS | Link Status | See bit description | RO |
| 54h–57h | SLCAP | Slot Capabilities Register | 00000060h | R/WO, RO |
| 58h–59h | SLCTL | Slot Control | 0000h | R/W, RO |
| 5Ah–5Bh | SLSTS | Slot Status | 0000h | R/WC, RO |
| 5Ch–5Dh | RCTL | Root Control | 0000h | R/W |
| 60h–63h | RSTS | Root Status | 00000000h | R/WC, RO |
| 80h–81h | MID | Message Signaled Interrupt Identifiers | 9005h | RO |
| 82h–83h | MC | Message Signaled Interrupt Message Control | 0000h | R/W, RO |
| 84h–87h | MA | Message Signaled Interrupt Message Address | 00000000h | R/W |
| 88h–89h | MD | Message Signaled Interrupt Message Data | 0000h | R/W |
| 90h–91h | SVCAP | Subsystem Vendor Capability | A00Dh | RO |
| 94h–97h | SVID | Subsystem Vendor Identification | 00000000h | R/WO |
| A0h–A1h | PMCAP | Power Management Capability | 0001h | RO |
| A2h–A3h | PMC | PCI Power Management Capability | C802h | RO |
| A4–A7h | PMCS | PCI Power Management Control and Status | 00000000h | R/W, RO |
| D8–DBh | MPC | Miscellaneous Port Configuration | 00110000h | R/W |
| DC–DFh | SMSCS | SMI/SCI Status | 00000000h | R/WC |
| E1h | RPDCGEN | Root Port Dynamic Clock Gating Enable (Netbook Only) | 00h | R/W |
| E2–E3h | IPWS | Intel® PRO/Wireless 3945ABG Status | 0000h | RO |
| 100–103h | VCH | Virtual Channel Capability Header | 18010002h | RO |
| 108h–10Bh | VCAP2 | Virtual Channel Capability 2 | 00000001h | RO |
| 10Ch–10Dh | PVC | Port Virtual Channel Control | 0000h | R/W |
| 10Eh–10Fh | PVS | Port Virtual Channel Status | 0000h | RO |
| 110h–113h | V0CAP | Virtual Channel 0 Resource Capability | 00000001h | RO |
| 114–117h | V0CTL | Virtual Channel 0 Resource Control | 800000FFh | R/W, RO |
| 11A–11Bh | V0STS | Virtual Channel 0 Resource Status | 0000h | RO |
| 144h–147h | UES | Uncorrectable Error Status | See bit description | R/WC, RO |
| 148h–14Bh | UEM | Uncorrectable Error Mask | 00000000h | R/WO, RO |
| 14Ch–14Fh | UEV | Uncorrectable Error Severity | 00060011h | RO |
| 150h–153h | CES | Correctable Error Status | 00000000h | R/WC |
| 154h–157h | CEM | Correctable Error Mask | 00000000h | R/WO |
| 158h–15Bh | AECC | Advanced Error Capabilities and Control | 00000000h | RO |
| 170h–173h | RES | Root Error Status | 00000000h | R/WC, RO |
| 180h–183h | RCTCL | Root Complex Topology Capability List | 00010005h | RO |
| 184h–187h | ESD | Element Self Description | See bit description | RO |

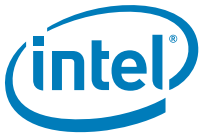


**Table 19-148.PCI Express* Configuration Registers Address Map (PCI Express—D28:F0/F1/F2/F3) (Sheet 3 of 3)**

| Offset | Mnemonic | Register Name | Function 0–5 Default | Type |
|---|---|---|---|---|
| 190h–193h | ULD | Upstream Link Description | 00000001h | RO |
| 198h–19Fh | ULBA | Upstream Link Base Address | See bit description | RO |
| 318h | PEETM | PCI Express Extended Test Mode Register | 00h | RO |

## 19.1.1 VID—Vendor Identification Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 00h–01h
Default Value: 8086h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | Vendor ID — RO. This is a 16-bit value assigned to Intel |

## 19.1.2 DID—Device Identification Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 02h–03h
Default Value: Port 1= Bit Description
Port 2= Bit Description
Port 3= Bit Description
Port 4= Bit Description
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Device ID** — RO. |

## 19.1.3 PCICMD—PCI Command Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 04h–05h
Default Value: 0000h
Attribute: R/W, RO
Size: 16 bits

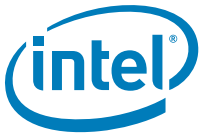
| Bit | Description |
|---|---|
| 15:11 | **Reserved** |
| 10 | **Interrupt Disable** — R/W. This bit disables pin-based INTx# interrupts on enabled Hot-Plug and power management events. This bit has no effect on MSI operation.<br>0 = Internal INTx# messages are generated if there is an interrupt for Hot-Plug or power management and MSI is not enabled.<br>1 = Internal INTx# messages will not be generated.<br><br>**NOTE:** This bit does not affect interrupt forwarding from devices connected to the root port. Assert_INTx and Deassert_INTx messages will still be forwarded to the internal interrupt controllers if this bit is set. |
| 9 | **Fast Back to Back Enable (FBE)** — **Reserved** per the *PCI Express* Base Specification*. |

| Bit | Description |
|---|---|
| 8 | **SERR# Enable (SEE)** — R/W.<br>0 = Disable.<br>1 = Enables the root port to generate an SERR# message when PSTS.SSE is set. |
| 7 | **Wait Cycle Control (WCC)** — **Reserved** per the *PCI Express Base Specification*. |
| 6 | **Parity Error Response (PER)** — R/W.<br>0 = Disable.<br>1 = Indicates that the device is capable of reporting parity errors as a master on the backbone. |
| 5 | VGA Palette Snoop (VPS) — **Reserved** per the *PCI Express* Base Specification*. |
| 4 | Postable Memory Write Enable (PMWE) — **Reserved** per the *PCI Express* Base Specification*. |
| 3 | Special Cycle Enable (SCE) — **Reserved** per the *PCI Express* Base Specification*. |
| 2 | **Bus Master Enable (BME)** — R/W.<br>0 = Disable. All cycles from the device are master aborted<br>1 = Enable. Allows the root port to forward cycles onto the backbone from a PCI Express* device. |
| 1 | **Memory Space Enable (MSE)** — R/W.<br>0 = Disable. Memory cycles within the range specified by the memory base and limit registers are master aborted on the backbone.<br>1 = Enable. Allows memory cycles within the range specified by the memory base and limit registers can be forwarded to the PCI Express device. |
| 0 | **I/O Space Enable (IOSE)** — R/W. This bit controls access to the I/O space registers.<br>0 = Disable. I/O cycles within the range specified by the I/O base and limit registers are master aborted on the backbone.<br>1 = Enable. Allows I/O cycles within the range specified by the I/O base and limit registers can be forwarded to the PCI Express device. |

### 19.1.4 PCISTS—PCI Status Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 06h–07h
Default Value: 0010h
Attribute: R/WC, RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15 | **Detected Parity Error (DPE)** — R/WC.<br>0 = **No** parity error detected.<br>1 = Root port received a command or data from the backbone with a parity error. This is set even if PCIMD.PER (D28:F0/F1/F2/F3:04, bit 6) is not set. |
| 14 | **Signaled System Error (SSE)** — R/WC.<br>0 = **No** system error signaled.<br>1 = Root port signaled a system error to the internal SERR# logic. |
| 13 | **Received Master Abort (RMA)** — R/WC.<br>0 = Root port has **not** received a completion with unsupported request status from the backbone.<br>1 = Root port received a completion with unsupported request status from the backbone. |



| Bit | Description |
|---|---|
| 12 | **Received Target Abort (RTA)** — R/WC.<br>0 = Root port has **not** received a completion with completer abort from the backbone.<br>1 = Root port received a completion with completer abort from the backbone. |
| 11 | **Signaled Target Abort (STA)** — R/WC.<br>0 = **No** target abort received.<br>1 = Root port forwarded a target abort received from the downstream device onto the backbone. |
| 10:9 | **DEVSEL# Timing Status (DEV_STS)** — **Reserved** per the *PCI Express* Base Specification*. |
| 8 | **Master Data Parity Error Detected (DPED)** — R/WC.<br>0 = **No** data parity error received.<br>1 = Root port received a completion with a data parity error on the backbone and PCIMD.PER (D28:F0/F1/F2/F3:04, bit 6) is set. |
| 7 | Fast Back to Back Capable (FB2BC) — **Reserved** per the *PCI Express* Base Specification*. |
| 6 | **Reserved** |
| 5 | **66 MHz Capable** — **Reserved** per the *PCI Express* Base Specification*. |
| 4 | Capabilities List — RO. Hardwired to 1. Indicates the presence of a capabilities list. |
| 3 | **Interrupt Status** — RO. Indicates status of Hot-Plug and power management interrupts on the root port that result in INTx# message generation.<br>0 = Interrupt is deasserted.<br>1 = Interrupt is asserted.<br>This bit is not set if MSI is enabled. If MSI is not enabled, this bit is set regardless of the state of PCICMD.Interrupt Disable bit (D28:F0/F1/F2/F3:04h:bit 10). |
| 2:0 | **Reserved** |

## 19.1.5 RID—Revision Identification Register (PCI Express—D28:F0/F1/F2/F3)

Offset Address: 08h — Attribute: RO
Default Value: See bit description — Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Revision ID** — RO. |

## 19.1.6 PI—Programming Interface Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 09h — Attribute: RO
Default Value: 00h — Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Programming Interface** — RO.<br>00h = No specific register level programming interface defined. |

### 19.1.7 SCC—Sub Class Code Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 0Ah
Default Value: 04h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Sub Class Code (SCC)** — RO.<br>04h = PCI-to-PCI bridge. |

### 19.1.8 BCC—Base Class Code Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 0Bh
Default Value: 06h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Base Class Code (BCC)** — RO.<br>06h = Indicates the device is a bridge device. |

### 19.1.9 CLS—Cache Line Size Register (PCI Express—D28:F0/F1/F2/F3)

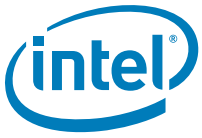
Address Offset: 0Ch
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Base Class Code (BCC)** — R/W. This is read/write but contains no functionality, per the *PCI Express* Base Specification*. |

### 19.1.10 PLT—Primary Latency Timer Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 0Dh
Default Value: 00h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:3 | Latency Count. **Reserved** per the *PCI Express* Base Specification.* |
| 2:0 | **Reserved** |



### 19.1.11 HEADTYP—Header Type Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 0Eh
Default Value: 81h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7 | **Multi-Function Device** — RO.<br>0 = Single-function device.<br>1 = Multi-function device. |
| 6:0 | Configuration Layout. Hardwired to 01h, which indicates a PCI-to-PCI bridge. |

### 19.1.12 BNUM—Bus Number Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 18–1Ah
Default Value: 000000h
Attribute: R/W
Size: 24 bits

| Bit | Description |
|---|---|
| 23:16 | **Subordinate Bus Number (SBBN)** — R/W. This field indicates the highest PCI bus number below the bridge. |
| 15:8 | **Secondary Bus Number (SCBN)** — R/W. This field indicates the bus number the port. |
| 7:0 | **Primary Bus Number (PBN)** — R/W. This field indicates the bus number of the backbone. |

### 19.1.13 IOBL—I/O Base and Limit Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 1Ch–1Dh
Default Value: 0000h
Attribute: R/W, RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:12 | **I/O Limit Address (IOLA)** — R/W. This field contains the I/O Base bits corresponding to address lines 15:12 for 4-KB alignment. Bits 11:0 are assumed to be padded to FFFh. |
| 11:8 | I/O Limit Address Capability (IOLC) — R/O. This field indicates that the bridge does not support 32-bit I/O addressing. |
| 7:4 | **I/O Base Address (IOBA)** — R/W. This field contains the I/O Base bits corresponding to address lines 15:12 for 4-KB alignment. Bits 11:0 are assumed to be padded to 000h. |
| 3:0 | **I/O Base Address Capability (IOBC)** — R/O. This field indicates that the bridge does not support 32-bit I/O addressing. |

## 19.1.14 SSTS—Secondary Status Register (PCI Express—D28:F0/F1/F2/F3)

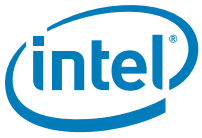
Address Offset: 1Eh–1Fh
Default Value: 0000h
Attribute: R/WC
Size: 16 bits

| Bit | Description |
|---|---|
| 15 | **Detected Parity Error (DPE)** — R/WC.<br>0 = **No** error.<br>1 = The port received a poisoned TLP. |
| 14 | **Received System Error (RSE)** — R/WC.<br>0 = **No** error.<br>1 = The port received an ERR_FATAL or ERR_NONFATAL message from the device. |
| 13 | **Received Master Abort (RMA)** — R/WC.<br>0 = Unsupported Request **not** received.<br>1 = The port received a completion with “Unsupported Request” status from the device. |
| 12 | **Received Target Abort (RTA)** — R/WC.<br>0 = Completion Abort **not** received.<br>1 = The port received a completion with “Completion Abort” status from the device. |
| 11 | **Signaled Target Abort (STA)** — R/WC.<br>0 = Completion Abort **not** sent.<br>1 = The port generated a completion with “Completion Abort” status to the device. |
| 10:9 | **Secondary DEVSEL# Timing Status (SDTS): Reserved** per *PCI Express* Base Specification*. |
| 8 | **Data Parity Error Detected (DPD)** — R/WC.<br>0 = Conditions below did **not** occur.<br>1 = Set when the BCTRL.PERE (D28:FO/F1/F2/F3:3E: bit 0) is set, and either of the following two conditions occurs:<br>— Port receives completion marked poisoned.<br>— Port poisons a write request to the secondary side. |
| 7 | **Secondary Fast Back to Back Capable (SFBC)**: **Reserved** per *PCI Express* Base Specification*. |
| 6 | **Reserved** |
| 5 | **Secondary 66 MHz Capable (SC66)**. **Reserved** per *PCI Express* Base Specification*. |
| 4:0 | **Reserved** |

## 19.1.15 MBL—Memory Base and Limit Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 20h–23h
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

Accesses that are within the ranges specified in this register will be sent to the attached device if CMD.MSE (D28:F0/F1/F2/F3:04:bit 1) is set. Accesses from the attached device that are outside the ranges specified will be forwarded to the backbone if CMD.BME (D28:F0/F1/F2/F3:04:bit 2) is set. The comparison performed is: MB ≥ AD[31:20] ≤ ML.



| Bit | Description |
|---|---|
| 31:20 | **Memory Limit (ML)** — R/W. These bits are compared with bits 31:20 of the incoming address to determine the upper 1-MB aligned value of the range. |
| 19:16 | **Reserved** |
| 15:4 | **Memory Base (MB)** — R/W. These bits are compared with bits 31:20 of the incoming address to determine the lower 1-MB aligned value of the range. |
| 3:0 | **Reserved** |

## 19.1.16 PMBL—Prefetchable Memory Base and Limit Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 24h–27h
Default Value: 00010001h
Attribute: R/W, RO
Size: 32 bits

Accesses that are within the ranges specified in this register will be sent to the device if CMD.MSE (D28:F0/F1/F2/F3;04, bit 1) is set. Accesses from the device that are outside the ranges specified will be forwarded to the backbone if CMD.BME (D28:F0/F1/F2/F3;04, bit 2) is set. The comparison performed is:
PMBU32:PMB ≥ AD[63:32]:AD[31:20] ≤ PMLU32:PML.

| Bit | Description |
|---|---|
| 31:20 | **Prefetchable Memory Limit (PML)** — R/W. These bits are compared with bits 31:20 of the incoming address to determine the upper 1-MB aligned value of the range. |
| 19:16 | 64-bit Indicator (I64L) — RO. This field indicates support for 64-bit addressing |
| 15:4 | **Prefetchable Memory Base (PMB)** — R/W. These bits are compared with bits 31:20 of the incoming address to determine the lower 1-MB aligned value of the range. |
| 3:0 | 64-bit Indicator (I64B) — RO. This field indicates support for 64-bit addressing |

## 19.1.17 PMBU32—Prefetchable Memory Base Upper 32 Bits Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 28h–2Bh
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Prefetchable Memory Base Upper Portion (PMBU)** — R/W. This field contains the Upper 32-bits of the prefetchable address base. |

## 19.1.18 PMLU32—Prefetchable Memory Limit Upper 32 Bits Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 2Ch–2Fh
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

| Bit | Description |
|---|---|
| 31:0 | **Prefetchable Memory Limit Upper Portion (PMLU)** — R/W. This field contains the Upper 32-bits of the prefetchable address limit. |

### 19.1.19 CAPP—Capabilities List Pointer Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 34h
Default Value: 40h
Attribute: R0
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Capabilities Pointer (PTR)** — RO. This field indicates that the pointer for the first entry in the capabilities list is at 40h in configuration space. |

### 19.1.20 INTR—Interrupt Information Register (PCI Express—D28:F0/F1/F2/F3)

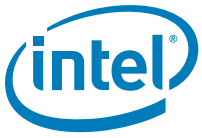
Address Offset: 3Ch–3Dh
Default Value: See bit description
Attribute: R/W, RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:8 | **Interrupt Pin (IPIN)** — RO. This field indicates the interrupt pin driven by the root port. At reset, this register takes on the following values that reflect the reset state of the D28IP register in chipset configuration space:<br><br>Port / Reset Value: 1 – D28IP.P1IP; 2 – D28IP.P2IP; 3 – D28IP.P3IP; 4 – D28IP.P4IP<br><br>**NOTE:** The value that is programmed into D28IP is always reflected in this register. |
| 7:0 | **Interrupt Line (ILINE)** — R/W. Default = 00h. This field is a software written value to indicate which interrupt line (vector) the interrupt is connected to. No hardware action is taken on this register. |

### 19.1.21 BCTRL—Bridge Control Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 3Eh–3Fh
Default Value: 0000h
Attribute: R/W
Size: 16 bits

| Bit | Description |
|---|---|
| 15:12 | **Reserved** |
| 11 | **Discard Timer SERR# Enable (DTSE)**. **Reserved** per *PCI Express* Base Specification*, Revision 1.0a |
| 10 | **Discard Timer Status (DTS)**. **Reserved** per *PCI Express* Base Specification*, Revision 1.0a. |
| 9 | **Secondary Discard Timer (SDT)**. **Reserved** per *PCI Express* Base Specification*, Revision 1.0a. |
| 8 | **Primary Discard Timer (PDT)**. **Reserved** per *PCI Express* Base Specification*, Revision 1.0a. |
| 7 | **Fast Back to Back Enable (FBE)**. **Reserved** per *PCI Express* Base Specification*, Revision 1.0a. |



| Bit | Description |
|---|---|
| 6 | **Secondary Bus Reset (SBR)** — R/W. This bit triggers a hot reset on the PCI Express* port. |
| 5 | Master Abort Mode (MAM): **Reserved** per Express specification. |
| 4 | **VGA 16-Bit Decode (V16)** — R/W.<br>0 = VGA range is enabled.<br>1 = The I/O aliases of the VGA range (see BCTRL:VE definition below), are not enabled, and only the base I/O ranges can be decoded |
| 3 | **VGA Enable (VE)**— R/W.<br>0 = The ranges below will **not** be claimed off the backbone by the root port.<br>1 = The following ranges will be claimed off the backbone by the root port:<br>— Memory ranges A0000h–BFFFFh<br>— I/O ranges 3B0h – 3BBh and 3C0h – 3DFh, and all aliases of bits 15:10 in any combination of 1s |
| 2 | **ISA Enable (IE)** — R/W. This bit only applies to I/O addresses that are enabled by the I/O Base and I/O Limit registers and are in the first 64 KB of PCI I/O space.<br>0 = The root port will **not** block any forwarding from the backbone as described below.<br>1 = The root port will block any forwarding from the backbone to the device of I/O transactions addressing the last 768 bytes in each 1-KB block (offsets 100h to 3FFh). |
| 1 | **SERR# Enable (SE)** — R/W.<br>0 = The messages described below are **not** forwarded to the backbone.<br>1 = ERR_COR, ERR_NONFATAL, and ERR_FATAL messages received are forwarded to the backbone. |
| 0 | **Parity Error Response Enable (PERE)** — R/W.<br>0 = Poisoned write TLPs and completions indicating poisoned TLPs will **not** set the SSTS.DPD (D28:F0/F1/F2/F3/F4/F5:1E, bit 8).<br>1 = Poisoned write TLPs and completions indicating poisoned TLPs will set the SSTS.DPD (D28:F0/F1/F2/F3/F4/F5:1E, bit 8). |

## 19.1.22 CLIST—Capabilities List Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 40–41h
Default Value: 8010h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:8 | **Next Capability (NEXT)** — RO. Value of 80h indicates the location of the next pointer. |
| 7:0 | **Capability ID (CID)** — RO. This field indicates this is a PCI Express* capability. |

### 19.1.23 XCAP—PCI Express* Capabilities Register (PCI Express—D28:F0/F1/F2/F3)

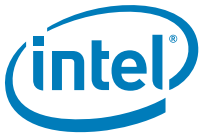
Address Offset: 42h–43h
Default Value: 0041h
Attribute: R/WO, RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:14 | **Reserved** |
| 13:9 | **Interrupt Message Number (IMN)** — RO. The Chipset does not have multiple MSI interrupt numbers. |
| 8 | **Slot Implemented (SI)** — R/WO. This field indicates whether the root port is connected to a slot. Slot support is platform specific. BIOS programs this field, and it is maintained until a platform reset. |
| 7:4 | **Device / Port Type (DT)** — RO. This field indicates this is a PCI Express* root port. |
| 3:0 | **Capability Version (CV)** — RO. This field indicates PCI Express 1.0. |

### 19.1.24 DCAP—Device Capabilities Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 44h–47h
Default Value: 00000FE0h
Attribute: RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:28 | **Reserved** |
| 27:26 | **Captured Slot Power Limit Scale (CSPS)** — RO. Not supported. |
| 25:18 | **Captured Slot Power Limit Value (CSPV)** — RO. Not supported. |
| 17:15 | **Reserved** |
| 14 | **Power Indicator Present (PIP)** — RO. This bit indicates no power indicator is present on the root port. |
| 13 | **Attention Indicator Present (AIP)** — RO. This bit indicates no attention indicator is present on the root port. |
| 12 | **Attention Button Present (ABP)** — RO. This bit indicates no attention button is present on the root port. |
| 11:9 | **Endpoint L1 Acceptable Latency (E1AL)** — RO. This field indicates more than 4 µs. This field essentially has no meaning for root ports since root ports are not endpoints. |
| 8:6 | **Endpoint L0 Acceptable Latency (E0AL)** — RO. This field indicates more than 64 µs. This field essentially has no meaning for root ports since root ports are not endpoints. |
| 5 | **Extended Tag Field Supported (ETFS)** — RO. This bit indicates that 8-bit tag fields are supported. |
| 4:3 | **Phantom Functions Supported (PFS)** — RO. This field indicates No phantom functions supported. |
| 2:0 | **Max Payload Size Supported (MPS)** — RO. This field indicates the maximum payload size supported is 128B. |



## 19.1.25 DCTL—Device Control Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 48h–49h
Default Value: 0000h
Attribute: R/W, RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15 | **Reserved** |
| 14:12 | **Max Read Request Size (MRRS)** — RO. Hardwired to 0. |
| 11 | **Enable No Snoop (ENS)** — RO. Not supported. The root port will not issue non-snoop requests. |
| 10 | **Aux Power PM Enable (APME)** — R/W. The OS will set this bit to 1 if the device connected has detected aux power. It has no effect on the root port otherwise. |
| 9 | **Phantom Functions Enable (PFE)** — RO. Not supported. |
| 8 | **Extended Tag Field Enable (ETFE)** — RO. Not supported. |
| 7:5 | **Max Payload Size (MPS)** — R/W. The root port only supports 128-B payloads, regardless of the programming of this field. |
| 4 | **Enable Relaxed Ordering (ERO)** — RO. Not supported. |
| 3 | **Unsupported Request Reporting Enable (URE)** — R/W.<br>0 = Disable. The root port will ignore unsupported request errors.<br>1 = Enable. The root port will generate errors when detecting an unsupported request. |
| 2 | **Fatal Error Reporting Enable (FEE)** — R/W.<br>0 = Disable. The root port will ignore fatal errors.<br>1 = Enable. The root port will generate errors when detecting a fatal error. |
| 1 | **Non-Fatal Error Reporting Enable (NFE)** — R/W.<br>0 = Disable. The root port will ignore non-fatal errors.<br>1 = Enable. The root port will generate errors when detecting a non-fatal error. |
| 0 | **Correctable Error Reporting Enable (CEE)** — R/W.<br>0 = Disable. The root port will ignore correctable errors.<br>1 = Enable. The root port will generate errors when detecting a correctable error. |

## 19.1.26 DSTS—Device Status Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 4Ah–4Bh
Default Value: 0010h
Attribute: R/WC, RO
Size: 16 bits

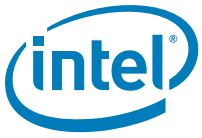
| Bit | Description |
|---|---|
| 15:6 | **Reserved** |
| 5 | **Transactions Pending (TDP)** — RO. This bit has no meaning for the root port since only one transaction may be pending to the Chipset, so a read of this bit cannot occur until it has already returned to 0. |
| 4 | **AUX Power Detected (APD)** — RO. The root port contains AUX power for wakeup. |
| 3 | **Unsupported Request Detected (URD)** — R/WC. This bit indicates an unsupported request was detected. |

| Bit | Description |
|---|---|
| 2 | **Fatal Error Detected (FED)** — R/WC. This bit indicates a fatal error was detected.<br>0 = Fatal has **not** occurred.<br>1 = A fatal error occurred from a data link protocol error, link training error, buffer overflow, or malformed TLP. |
| 1 | **Non-Fatal Error Detected (NFED)** — R/WC. This bit indicates a non-fatal error was detected.<br>0 = Non-fatal has **not** occurred.<br>1 = A non-fatal error occurred from a poisoned TLP, unexpected completions, unsupported requests, completer abort, or completer timeout. |
| 0 | **Correctable Error Detected (CED)** — R/WC. This bit indicates a correctable error was detected.<br>0 = Correctable has **not** occurred.<br>1 = The port received an internal correctable error from receiver errors / framing errors, TLP CRC error, DLLP CRC error, replay num rollover, replay timeout. |

## 19.1.27 LCAP—Link Capabilities Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 4Ch–4Fh  Attribute: R/W, RO
Default Value: See bit description  Size: 32 bits

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:24</td><td>Port Number (PN) — RO. This field indicates the port number for the root port. This value is different for each implemented port:
<table>
<tr><th>Function</th><th>Port #</th><th>Value of PN Field</th></tr>
<tr><td>D28:F0</td><td>1</td><td>01h</td></tr>
<tr><td>D28:F1</td><td>2</td><td>02h</td></tr>
<tr><td>D28:F2</td><td>3</td><td>03h</td></tr>
<tr><td>D28:F3</td><td>4</td><td>04h</td></tr>
</table></td></tr>
<tr><td>23:21</td><td>Reserved</td></tr>
<tr><td>20</td><td>Link Active Reporting Capable (LARC) — RO. Hardwired to 1 to indicate that this port supports the optional capability of reporting the DL_Active state of the Data Link Control and Management State Machine.</td></tr>
<tr><td>19:18</td><td>Reserved</td></tr>
<tr><td>17:15</td><td>L1 Exit Latency (EL1) — RO. Set to 010b to indicate an exit latency of 2 µs to 4 µs.</td></tr>
<tr><td>14:12</td><td>L0s Exit Latency (EL0) — RO. This field indicates as exit latency based upon common-clock configuration.<br><br>LCLT.CCC — Value of EL0 (these bits)<br>0 — MPC.UCEL (D28:F0/F1/F2/F3:D8h:bits20:18)<br>1 — MPC.CCEL (D28:F0/F1/F2/F3:D8h:bits17:15)<br><br>NOTE: LCLT.CCC is at D28:F0/F1/F2/F3:50h:bit 6</td></tr>
</table>



| Bit | Description |
|---|---|
| 11:10 | **Active State Link PM Support (APMS)** — R/WO. This field indicates what level of active state link power management is supported on the root port.<br><br>Bits: 00b — Neither L0s nor L1 are supported<br>Bits: 01b — L0s Entry Supported<br>Bits: 10b — L1 Entry Supported<br>Bits: 11b — Both L0s and L1 Entry Supported |
| 9:4 | **Maximum Link Width (MLW)** — RO. For the root ports, several values can be taken, based upon the value of the chipset configuration register field RPC.PC1 (Chipset Configuration Registers:Offset 0224h:bits1:0) for Ports 1–4.<br><br>Port # 1: RPC.PC1=00b: 01h; RPC.PC1=11b: 04h<br>Port # 2: RPC.PC1=00b: 01h; RPC.PC1=11b: 01h<br>Port # 3: RPC.PC1=00b: 01h; RPC.PC1=11b: 01h<br>Port # 4: RPC.PC1=00b: 01h; RPC.PC1=11b: 01h |
| 3:0 | **Maximum Link Speed (MLS)** — RO. Set to 1h to indicate the link speed is 2.5 Gb/s. |

## 19.1.28 LCTL—Link Control Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 50h-51h
Default Value: 0000h
Attribute: R/W, WO, RO
Size: 16 bits

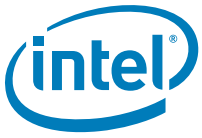
| Bit | Description |
|---|---|
| 15:8 | **Reserved** |
| 7 | **Extended Synch (ES)** — R/W.<br>0 = Extended synch disabled.<br>1 = Forces extended transmission of FTS ordered sets in FTS and extra TS2 at exit from L1 prior to entering L0. |
| 6 | **Common Clock Configuration (CCC)** — R/W.<br>0 = The Chipset and device are **not** using a common reference clock.<br>1 = The Chipset and device are operating with a distributed common reference clock. |
| 5 | **Retrain Link (RL)** — WO.<br>0 = This bit always returns 0 when read.<br>1 = The root port will train its downstream link.<br><br>**NOTE:** Software uses LSTS.LT (D28:F0/F1/F2/F3/F4/F5:52, bit 11) to check the status of training. |
| 4 | **Link Disable (LD)** — R/W.<br>0 = Link enabled.<br>1 = The root port will disable the link. |
| 3 | **Read Completion Boundary Control (RCBC)** — RO. This bit indicates that the read completion boundary is 64 bytes. |
| 2 | **Reserved** |

| Bit | Description |
|---|---|
| 1:0 | **Active State Link PM Control (APMC)** — R/W. This bit indicates whether the root port should enter L0s or L1 or both.<br>00b = Disabled<br>01b = L0s Entry is Enabled<br>10b = L1 Entry is Enabled<br>11b = L0s and L1 Entry Enabled |

## 19.1.29 LSTS—Link Status Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 52h–53h Attribute: RO
Default Value: See bit description Size: 16 bits

| Bit | Description |
|---|---|
| 15:14 | **Reserved** |
| 13 | **Data Link Layer Active (DLLA)** — RO. D<br>0 = Data Link Control and Management State Machine is not in the DL_Active state. (Default)<br>1 = Data Link Control and Management State Machine is in the DL_Active state. |
| 12 | **Slot Clock Configuration (SCC)** — RO. Set to 1b to indicate that the Chipset uses the same reference clock as on the platform and does not generate its own clock. |
| 11 | **Link Training (LT)** — RO.<br>0 = Link training completed. (Default)<br>1 = Link training is occurring. |
| 10 | Link Training Error (LTE) — RO. Not supported. Set value is 0b. |
| 9:4 | **Negotiated Link Width (NLW)** — RO. This field indicates the negotiated width of the given PCI Express* link. The contents of this NLW field is undefined if the link has not successfully trained.<br>Port # / Possible Values:<br>1 — 000001b, 000010b, 000100b<br>2 — 000001b<br>3 — 000001b<br>4 — 000001b<br>**NOTE:** 000001b = x1 link width, 000010b =x2 linkwidth (not supported), 000100 = x4 linkwidth |
| 3:0 | **Link Speed (LS)** — RO. This field indicates the negotiated Link speed of the given PCI Express* link.<br>01h = Link is 2.5 Gb/s. |



## 19.1.30 SLCAP—Slot Capabilities Register (PCI Express—D28:F0/F1/F2/F3)

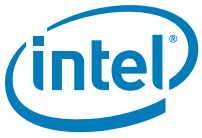
Address Offset: 54h–57h Attribute: R/WO, RO
Default Value: 00000060h Size: 32 bits

| Bit | Description |
|---|---|
| 31:19 | **Physical Slot Number (PSN)** — R/WO. This is a value that is unique to the slot number. BIOS sets this field and it remains set until a platform reset. |
| 18:17 | **Reserved** |
| 16:15 | **Slot Power Limit Scale (SLS)** — R/WO. This field specifies the scale used for the slot power limit value. BIOS sets this field and it remains set until a platform reset.<br>Range of Values:<br>00b = 1.0 x<br>01b = 0.1 x<br>10b = 0.01 x<br>11b = 0.001 x |
| 14:7 | **Slot Power Limit Value (SLV)** — R/WO. This field specifies the upper limit (in conjunction with SLS value), on the upper limit on power supplied by the slot. The two values together indicate the amount of power in watts allowed for the slot. Power limit (in Watts) is calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field. BIOS sets this field and it remains set until a platform reset. |
| 6 | **Hot Plug Capable (HPC)** — RO.<br>1b = Hot-Plug is supported. |
| 5 | **Hot Plug Surprise (HPS)** — RO.<br>1b = Device may be removed from the slot without prior notification. |
| 4 | **Power Indicator Present (PIP)** — RO.<br>0b = Power indicator LED is not present for this slot. |
| 3 | **Attention Indicator Present (AIP)** — RO.<br>0b = Attention indicator LED is not present for this slot. |
| 2 | **MRL Sensor Present (MSP)** — RO.<br>0b = MRL sensor is not present. |
| 1 | **Power Controller Present (PCP)** — RO.<br>0b = Power controller is not implemented for this slot. |
| 0 | **Attention Button Present (ABP)** — RO.<br>0b =Attention button is not implemented for this slot. |

### 19.1.31 SLCTL—Slot Control Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 58h–59h
Default Value: 0000h
Attribute: R/W, RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:13 | **Reserved** |
| 12 | **Link Active Changed Enable (LACE)** — RW.<br>0 = Disable.<br>1 = Enables generation of a hot plug interrupt when the Data Link Layer Link Active field (D28:F0/F1/F2/F3/F4/F5:52h:bit 13) is changed. |
| 11 | **Reserved** |
| 10 | **Power Controller Control (PCC)** — RO.This bit has no meaning for module based Hot-Plug. |
| 9:8 | **Power Indicator Control (PIC)** — R/W. When read, the current state of the power indicator is returned. When written, the appropriate POWER_INDICATOR_* messages are sent. Defined encodings are:<br><br>Bits / Definition<br>00b — **Reserved**<br>01b — On<br>10b — Blink<br>11b — Off |
| 7:6 | **Attention Indicator Control (AIC)** — R/W. When read, the current state of the attention indicator is returned. When written, the appropriate ATTENTION_INDICATOR_* messages are sent. Defined encodings are:<br><br>Bits / Definition<br>00b — **Reserved**<br>01b — On<br>10b — Blink<br>11b — Off |
| 5 | **Hot Plug Interrupt Enable (HPE)** — R/W.<br>0 = Disable. Hot plug interrupts based on Hot-Plug events is disabled.<br>1 = Enables generation of a Hot-Plug interrupt on enabled Hot-Plug events. |
| 4 | **Command Completed Interrupt Enable (CCE)** — R/W.<br>0 = Disable. Hot plug interrupts based on command completions is disabled.<br>1 = Enables the generation of a Hot-Plug interrupt when a command is completed by the Hot-Plug controller. |
| 3 | **Presence Detect Changed Enable (PDE)** — R/W.<br>0 = Disable. Hot plug interrupts based on presence detect logic changes is disabled.<br>1 = Enables the generation of a Hot-Plug interrupt or wake message when the presence detect logic changes state. |
| 2 | **MRL Sensor Changed Enable (MSE)** — R/W. MSE not supported. |
| 1 | **Power Fault Detected Enable (PFE)** — R/W. PFE not supported. |



| Bit | Description |
|---|---|
| 0 | **Attention Button Pressed Enable (ABE)** — R/W. When set, enables the generation of a Hot-Plug interrupt when the attention button is pressed.<br>0 = Disable. Hot plug interrupts based on the attention button being pressed is disabled.<br>1 = Enables the generation of a Hot-Plug interrupt when the attention button is pressed. |

## 19.1.32 SLSTS—Slot Status Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 5Ah–5Bh
Default Value: 0000h
Attribute: R/WC, RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:9 | **Reserved** |
| 8 | **Link Active State Changed (LASC)** — R/WC.<br>0 = No Change.<br>1 = Value reported in Data Link Layer Link Active field of the Link Status register (D28:F0/F1/F2/F3/F4/F5:52h:bit 13) is changed. In response to a Data Link Layer State Changed event, software must read Data Link Layer Link Active field of the Link Status register to determine if the link is active before initiating configuration cycles to the hot plugged device. |
| 7 | **Reserved** |
| 6 | **Presence Detect State (PDS)** — RO. If XCAP.SI (D28:F0/F1/F2/F3/F4/F5:42h:bit 8) is set indicating that this root port spawns a slot), then this bit:<br>0 = Slot is empty.<br>1 = Slot has a device connected.<br>Otherwise; if XCAP.SI is cleared, this bit is always set to 1. |
| 5 | **MRL Sensor State (MS)** — **Reserved** as the MRL sensor is not implemented. |
| 4 | **Command Completed (CC)** — R/WC.<br>0 = Issued command **not** completed.<br>1 = The Hot-Plug controller completed an issued command. This is set when the last message of a command is sent and indicates that software can write a new command to the slot controller. |
| 3 | **Presence Detect Changed (PDC)** — R/WC.<br>0 = **No** change in the PDS bit.<br>1 = The PDS bit changed states. |
| 2 | **MRL Sensor Changed (MSC)** — **Reserved** as the MRL sensor is not implemented. |
| 1 | **Power Fault Detected (PFD)** — **Reserved** as a power controller is not implemented. |
| 0 | **Attention Button Pressed (ABP)** — R/WC.<br>0 = The attention button has **not** been pressed.<br>1 = The attention button is pressed. |

### 19.1.33 RCTL—Root Control Register (PCI Express—D28:F0/F1/F2/F3)

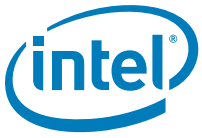
Address Offset: 5Ch–5Dh
Default Value: 0000h
Attribute: R/W
Size: 16 bits

| Bit | Description |
|---|---|
| 15:4 | **Reserved** |
| 3 | **PME Interrupt Enable (PIE)** — R/W.<br>0 = Interrupt generation disabled.<br>1 = Interrupt generation enabled when PCISTS.Interrupt Status (D28:F0/F1/F2/F3/F4/F5:60h, bit 16) is in a set state (either due to a 0 to 1 transition, or due to this bit being set with RSTS.IS already set). |
| 2 | **System Error on Fatal Error Enable (SFE)** — R/W.<br>0 = Disable. An SERR# will **not** be generated.<br>1 = Enable. An SERR# will be generated, assuming CMD.SEE (D28:F0/F1/F2/F3/F4/F5:04, bit 8) is set, if a fatal error is reported by any of the devices in the hierarchy of this root port, including fatal errors in this root port. |
| 1 | **System Error on Non-Fatal Error Enable (SNE)** — R/W.<br>0 = Disable. An SERR# will **not** be generated.<br>1 = Enable. An SERR# will be generated, assuming CMD.SEE (D28:F0/F1/F2/F3/F4/F5:04, bit 8) is set, if a non-fatal error is reported by any of the devices in the hierarchy of this root port, including non-fatal errors in this root port. |
| 0 | **System Error on Correctable Error Enable (SCE)** — R/W.<br>0 = Disable. An SERR# will **not** be generated.<br>1 = Enable. An SERR# will be generated, assuming CMD.SEE (D28:F0/F1/F2/F3/F4/F5:04, bit 8) if a correctable error is reported by any of the devices in the hierarchy of this root port, including correctable errors in this root port. |

### 19.1.34 RSTS—Root Status Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 60h–63h
Default Value: 00000000h
Attribute: R/WC, RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:18 | **Reserved** |
| 17 | **PME Pending (PP)** — RO.<br>0 = Indicates no more PMEs are pending.<br>1 = Indicates another PME is pending (this is implicit because of the definition of this bit being 1). Hardware will set the PME Status bit again and update the Requestor ID appropriately. The PME pending bit is cleared by hardware if no more PMEs are pending. |
| 16 | **PME Status (PS)** — R/WC.<br>0 = PME was **not** asserted.<br>1 = PME was asserted by the requestor ID in RID. Subsequent PMEs are kept pending until this bit is cleared. |
| 15:0 | **PME Requestor ID (RID)** — RO. This field indicates the PCI requestor ID of the last PME requestor. The value in this field is valid only when PS is set. |



## 19.1.35 MID—Message Signaled Interrupt Identifiers Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 80h–81h
Default Value: 9005h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:8 | **Next Pointer (NEXT)** — RO. This field indicates the location of the next pointer in the list. |
| 7:0 | **Capability ID (CID)** — RO. Capabilities ID indicates MSI. |

## 19.1.36 MC—Message Signaled Interrupt Message Control Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 82–83h
Default Value: 0000h
Attribute: R/W, RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:8 | **Reserved** |
| 7 | **64 Bit Address Capable (C64)** — RO. Capable of generating a 32-bit message only. |
| 6:4 | **Multiple Message Enable (MME)** — R/W. These bits are R/W for software compatibility, but only one message is ever sent by the root port. |
| 3:1 | **Multiple Message Capable (MMC)** — RO. Only one message is required. |
| 0 | **MSI Enable (MSIE)** — R/W.<br>0 = Disabled.<br>1 = Enabled and traditional interrupt pins are not used to generate interrupts.<br><br>**NOTE:** CMD.BME (D28:F0/F1/F2/F3:04h:bit 2) must be set for an MSI to be generated. If CMD.BME is cleared, and this bit is set, no interrupts (not even pin based) are generated. |

## 19.1.37 MA—Message Signaled Interrupt Message Address Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 84h–87h
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

| Bit | Description |
|---|---|
| 31:2 | **Address (ADDR)** — R/W. This field contains the lower 32 bits of the system specified message address; always DWord aligned. |
| 1:0 | **Reserved** |

### 19.1.38 MD—Message Signaled Interrupt Message Data Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 88h–89h
Default Value: 0000h
Attribute: R/W
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | **Data (DATA)** — R/W. This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the lower word (PCI AD[15:0]) during the data phase of the MSI memory write transaction. |

### 19.1.39 SVCAP—Subsystem Vendor Capability Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 90h–91h
Default Value: A00Dh
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:8 | **Next Capability (NEXT)** — RO. This field indicates the location of the next pointer in the list. |
| 7:0 | **Capability Identifier (CID)** — RO. Value of 0Dh indicates this is a PCI bridge subsystem vendor capability. |

### 19.1.40 SVID—Subsystem Vendor Identification Register (PCI Express—D28:F0/F1/F2/F3)

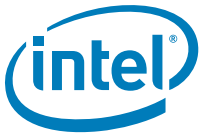
Address Offset: 94h–97h
Default Value: 00000000h
Attribute: R/WO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:16 | **Subsystem Identifier (SID)** — R/WO. This field indicates the subsystem as identified by the vendor. This field is write once and is locked down until a bridge reset occurs (not the PCI bus reset). |
| 15:0 | **Subsystem Vendor Identifier (SVID)** — R/WO. This field indicates the manufacturer of the subsystem. This field is write once and is locked down until a bridge reset occurs (not the PCI bus reset). |

### 19.1.41 PMCAP—Power Management Capability Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: A0h–A1h
Default Value: 0001h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:8 | **Next Capability (NEXT)** — RO. This field indicates that this is the last item in the list. |
| 7:0 | **Capability Identifier (CID)** — RO. Value of 01h indicates this is a PCI power management capability. |



## 19.1.42 PMC—PCI Power Management Capabilities Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: A2h–A3h
Default Value: C802h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:11 | **PME_Support (PMES)** — RO. This field indicates PME# is supported for states D0, $D3_{HOT}$ and $D3_{COLD}$. The root port does not generate PME#, but reporting that it does is necessary for some legacy operating systems to enable PME# in devices connected behind this root port. |
| 10 | **D2_Support (D2S)** — RO. The D2 state is not supported. |
| 9 | **D1_Support (D1S)** — RO The D1 state is not supported. |
| 8:6 | **Aux_Current (AC)** — RO. This field reports 375 mA maximum suspend well current required when in the $D3_{COLD}$ state. |
| 5 | **Device Specific Initialization (DSI)** — RO. This bit indicates that no device-specific initialization is required. |
| 4 | **Reserved** |
| 3 | **PME Clock (PMEC)** — RO. This bit indicates that PCI clock is not required to generate PME#. |
| 2:0 | **Version (VS)** — RO. This field indicates support for *Revision 1.1 of the PCI Power Management Specification*. |

## 19.1.43 PMCS—PCI Power Management Control and Status Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: A4h–A7h
Default Value: 00000000h
Attribute: R/W, RO
Size: 32 bits

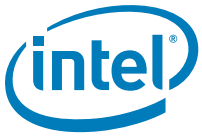
| Bit | Description |
|---|---|
| 31:24 | **Reserved** |
| 23 | **Bus Power / Clock Control Enable (BPCE)**. **Reserved** per *PCI Express* Base Specification, Revision 1.0a*. |
| 22 | **B2/B3 Support (B23S)**. **Reserved** per *PCI Express* Base Specification, Revision 1.0a*. |
| 21:16 | **Reserved** |
| 15 | **PME Status (PMES)** — RO. This bit indicates a PME was received on the downstream link. |
| 14:9 | **Reserved** |
| 8 | **PME Enable (PMEE)** — R/W. Indicates PME is enabled. The root port takes no action on this bit, but it must be R/W for some legacy operating systems to enable PME# on devices connected to this root port.<br>0 = Disable.<br>1 = Enable.<br><br>**NOTE:** This bit is sticky and resides in the resume well. The reset for this bit is RSMRST# which is not asserted during a warm reset. |
| 7:2 | **Reserved** |

| Bit | Description |
|---|---|
| 1:0 | **Power State (PS)** — R/W. This field is used both to determine the current power state of the root port and to set a new power state. The values are:<br>00 = D0 state<br>11 = $D3_{HOT}$ state<br><br>**NOTE:** When in the $D3_{HOT}$ state, the controller's configuration space is available, but the I/O and memory spaces are not. Type 1 configuration cycles are also not accepted. Interrupts are not required to be blocked as software will disable interrupts prior to placing the port into D3HOT. If software attempts to write a '10' or '01' to these bits, the write will be ignored. |

## 19.1.44 MPC—Miscellaneous Port Configuration Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: D8h–DBh Attribute: R/W
Default Value: 00110000h Size: 32 bits

| Bit | Description |
|---|---|
| 31 | **Power Management SCI Enable (PMCE)** — R/W.<br>0 = Disable. SCI generation based on a power management event is disabled.<br>1 = Enables the root port to generate SCI whenever a power management event is detected. |
| 30 | **Hot Plug SCI Enable (HPCE)** — R/W.<br>0 = Disable. SCI generation based on a Hot-Plug event is disabled.<br>1 = Enables the root port to generate SCI whenever a Hot-Plug event is detected. |
| 29 | **Link Hold Off (LHO)**— R/W.<br>0 = **Not** in Link Hold Off.<br>1 = The port will not take any TLP. This is used during loopback mode to fill up the downstream queue. |
| 28 | **Address Translator Enable (ATE)** — R/W. This bit is used to enable address translation via the AT bits in this register during loopback mode.<br>0 = Disable.<br>1 = Enable. |
| 27 | **Reserved**. |
| 26 | **Invalid Receive Bus Number Check Enable (IRBNCE)** — R/W.<br>0 = Disable.<br>1 = Enable. Receive transaction layer will signal an error if the bus number of a Memory request does not fall within the range between SCBN and SBBN. If this check is enabled and the request is a memory write, it is treated as an Unsupported Request. If this check is enabled and the request is a non-posted memory read request, the request is considered a Malformed TLP and a fatal error.<br><br>**NOTE:** Messages, IO, Configuration, and Completions are not checked for valid bus number. |

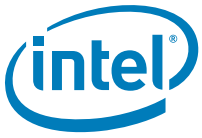


| Bit | Description |
|---|---|
| 25 | **Invalid Receive Range Check Enable (IRRCE)** — R/W.<br>0 = Disable.<br>1 = Enable. Receive transaction layer will treat the TLP as an Unsupported Request error if the address range of a memory request does not outside the range between prefetchable and non-prefetchable base and limit.<br><br>**NOTE:** Messages, I/O, Configuration, and Completions are not checked for valid address ranges. |
| 24 | **BME Receive Check Enable (BMERCE)** — R/W.<br>0 = Disable.<br>1 = Enable. Receive transaction layer will treat the TLP as an Unsupported Request error if a memory read or write request is received and the Bus Master Enable bit is not set.<br><br>**NOTE:** Messages, IO, Configuration, and Completions are not checked for BME. |
| 23:21 | **Reserved** |
| 20:18 | **Unique Clock Exit Latency (UCEL)** — R/W. This value represents the L0s Exit Latency for unique-clock configurations (LCTL.CCC = 0) (D28:F0/F1/F2/F3/F4/ F5:Offset 50h:bit 6). It defaults to 512 ns to less than 1 µs, but may be overridden by BIOS. |
| 17:15 | **Common Clock Exit Latency (CCEL)** — R/W. This value represents the L0s Exit Latency for common-clock configurations (LCTL.CCC = 1) (D28:F0/F1/F2/F3:Offset 50h:bit 6). It defaults to 128 ns to less than 256 ns, but may be overridden by BIOS. |
| 14:8 | **Reserved** |
| 7 | **Port I/OxApic Enable (PAE)** — R/W.<br>0 = Hole is disabled.<br>1 = A range is opened through the bridge for the following memory addresses:<br><br>Port # / Address:<br>1 — FEC1_0000h – FEC1_7FFFh<br>2 — FEC1_8000h – FEC1_FFFFh<br>3 — FEC2_0000h – FEC2_7FFFh<br>4 — FEC2_8000h – FEC2_FFFFh |
| 6:2 | **Reserved** |
| 1 | **Hot Plug SMI Enable (HPME)** — R/W.<br>0 = Disable. SMI generation based on a Hot-Plug event is disabled.<br>1 = Enables the root port to generate SMI whenever a Hot-Plug event is detected. |
| 0 | **Power Management SMI Enable (PMME)** — R/W.<br>0 = Disable. SMI generation based on a power management event is disabled.<br>1 = Enables the root port to generate SMI whenever a power management event is detected. |

### 19.1.45 SMSCS—SMI/SCI Status Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: DCh–DFh Attribute: R/WC
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31 | **Power Management SCI Status (PMCS)** — R/WC.<br>0 = Interrupt **Not** needed.<br>1 = PME control logic needs to generate an interrupt, and this interrupt has been routed to generate an SCI. |
| 30 | **Hot Plug SCI Status (HPCS)** — R/WC.<br>0 = Interrupt **Not** needed.<br>1 = Hot-Plug controller needs to generate an interrupt, and has this interrupt been routed to generate an SCI. |
| 29:5 | **Reserved** |
| 4 | **Hot Plug Link Active State Changed SMI Status (HPLAS)** — R/WC.<br>0 = No change<br>1 = SLSTS.LASC (D28:F0/F1/F2/F3:5A, bit 8) transitioned from 0-to-1, and MPC.HPME (D28:F0/F1/F2/F3:D8, bit 1) is set. When this bit is set, an SMI# will be generated. |
| 3 | **Hot Plug Command Completed SMI Status (HPCCM)** — R/WC.<br>0 = **No** change<br>1 = SLSTS.CC (D28:F0/F1/F2/F3/:5A, bit 4) transitioned from 0-to-1, and MPC.HPME (D28:F0/F1/F2/F3:D8, bit 1) is set. When this bit is set, an SMI# will be generated. |
| 2 | **Hot Plug Attention Button SMI Status (HPABM)** — R/WC.<br>0 = **No** change<br>1 = SLSTS.ABP (D28:F0/F1/F2/F3:5A, bit 0) transitioned from 0-to-1, and MPC.HPME (D28:F0/F1/F2/F3:D8, bit 1) is set. When this bit is set, an SMI# will be generated. |
| 1 | **Hot Plug Presence Detect SMI Status (HPPDM)** — R/WC.<br>0 = **No** change<br>1 = SLSTS.PDC (D28:F0/F1/F2/F3:5A, bit 3) transitions from 0-to-1, and MPC.HPME (D28:F0/F1/F2/F3:D8, bit 1) is set. When this bit is set, an SMI# will be generated. |
| 0 | **Power Management SMI Status (PMMS)** — R/WC.<br>0 = **No** change<br>1 = RSTS.PS (D28:F0/F1/F2/F3:60, bit 16) transitions from 0-to-1, and MPC.PMME (D28:F0/F1/F2/F3:D8, bit 1) is set. |



## 19.1.46 RPDCGEN - Root Port Dynamic Clock Gating Enable (PCI Express-D28:F0/F1/F2/F3) (Netbook Only)

Address Offset: E1h
Default Value: 00h
Attribute: R/W
Size: 8-bits

| Bits | Description |
|---|---|
| 7:4 | **Reserved**. RO |
| 3 | **Shared Resource Dynamic Link Clock Gating Enable (SRDLCGEN) —** RW.<br>0 = Disables dynamic clock gating of the shared resource link clock domain.<br>1 = Enables dynamic clock gating on the root port shared resource link clock domain.<br>Only the value from Port 1 is used for ports 1–4. |
| 2 | **Shared Resource Dynamic Backbone Clock Gate Enable (SRDBCGEN) —** RW.<br>0 = Disables dynamic clock gating of the shared resource backbone clock domain.<br>1 = Enables dynamic clock gating on the root port shared resource backbone clock domain.<br>Only the value from Port 1 is used for ports 1–4. |
| 1 | **Root Port Dynamic Link Clock Gate Enable (RPDLCGEN) —** RW.<br>0 = Disables dynamic clock gating of the root port link clock domain.<br>1 = Enables dynamic clock gating on the root port link clock domain. |
| 0 | **Root Port Dynamic Backbone Clock Gate Enable (RPDBCGEN) —** RW.<br>0 = Disables dynamic clock gating of the root port backbone clock domain.<br>1 = Enables dynamic clock gating on the root port backbone clock domain. |

## 19.1.47 IPWS—Intel® PRO/Wireless 3945ABG Status (PCI Express—D28:F0/F1/F2/F3) (Netbook Only)

Address Offset: E2h–E3h
Default Value: 0007h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15 | **Intel PRO/Wireless 3945ABG Status (IPWSTAT)** — RO. This bit is set if the link has trained to L0 in Intel PRO/Wireless 3945ABG mode. |
| 14:0 | **Reserved** |

## 19.1.48 VCH—Virtual Channel Capability Header Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 100h–103h
Default Value: 18010002h
Attribute: RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:20 | **Next Capability Offset (NCO)** — RO. This field indicates the next item in the list. |
| 19:16 | **Capability Version (CV)** — RO. This field indicates that this is version 1 of the capability structure by the PCI SIG. |
| 15:0 | **Capability ID (CID)** — RO. This field indicates that this is the Virtual Channel capability item. |

### 19.1.49 VCAP2—Virtual Channel Capability 2 Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 108h–10Bh
Default Value: 00000001h
Attribute: RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:24 | **VC Arbitration Table Offset (ATO)** — RO. This field indicates that no table is present for VC arbitration since it is fixed. |
| 23:0 | **Reserved**. |

### 19.1.50 PVC—Port Virtual Channel Control Register (PCI Express—D28:F0/F1/F2/F3)

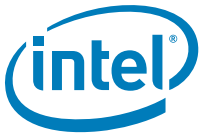
Address Offset: 10Ch–10Dh
Default Value: 0000h
Attribute: R/W
Size: 16 bits

| Bit | Description |
|---|---|
| 15:4 | **Reserved**. |
| 3:1 | **VC Arbitration Select (AS)** — R/W. This field indicates which VC should be programmed in the VC arbitration table. The root port takes no action on the setting of this field since there is no arbitration table. |
| 0 | **Load VC Arbitration Table (LAT)** — R/W. This bit indicates that the table programmed should be loaded into the VC arbitration table. This bit always returns 0 when read. |

### 19.1.51 PVS — Port Virtual Channel Status Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 10Eh–10Fh
Default Value: 0000h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:1 | **Reserved**. |
| 0 | **VC Arbitration Table Status (VAS)** — RO. This bit indicates the coherency status of the VC Arbitration table when it is being updated. This field is always 0 in the root port since there is no VC arbitration table. |



## 19.1.52 V0CAP — Virtual Channel 0 Resource Capability Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 110h–113h
Default Value: 00000001h
Attribute: RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:24 | **Port Arbitration Table Offset (AT)** — RO. This VC implements no port arbitration table since the arbitration is fixed. |
| 23 | **Reserved**. |
| 22:16 | **Maximum Time Slots (MTS)** — RO. This VC implements fixed arbitration; therefore, this field is not used. |
| 15 | **Reject Snoop Transactions (RTS)** — RO. This VC must be able to take snoopable transactions. |
| 14 | **Advanced Packet Switching (APS)** — RO. This VC is capable of all transactions, not just advanced packet switching transactions. |
| 13:8 | **Reserved**. |
| 7:0 | **Port Arbitration Capability (PAC)** — RO. This field indicates that this VC uses fixed port arbitration. |

## 19.1.53 V0CTL — Virtual Channel 0 Resource Control Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 114h–117h
Default Value: 800000FFh
Attribute: R/W, RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31 | **Virtual Channel Enable (EN**) — RO. Always set to 1. Virtual Channel 0 cannot be disabled. |
| 30:27 | **Reserved**. |
| 26:24 | **Virtual Channel Identifier (VCID)** — RO. This field indicates the ID to use for this virtual channel. |
| 23:20 | **Reserved**. |
| 19:17 | **Port Arbitration Select (PAS)** — R/W. This field indicates which port table is being programmed. The root complex takes no action on this setting since the arbitration is fixed and there is no arbitration table. |
| 16 | **Load Port Arbitration Table (LAT)** — RO. The root port does not implement an arbitration table for this virtual channel. |
| 15:8 | **Reserved**. |

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7:1</td><td>Transaction Class / Virtual Channel Map (TVM) — R/W. This field indicates which transaction classes are mapped to this virtual channel. When a bit is set, this transaction class is mapped to the virtual channel.<br><br>
<table>
<tr><th>Bit</th><th>Transaction Class</th></tr>
<tr><td>7</td><td>Transaction Class 7</td></tr>
<tr><td>6</td><td>Transaction Class 6</td></tr>
<tr><td>5</td><td>Transaction Class 5</td></tr>
<tr><td>4</td><td>Transaction Class 4</td></tr>
<tr><td>3</td><td>Transaction Class 3</td></tr>
<tr><td>2</td><td>Transaction Class 2</td></tr>
<tr><td>1</td><td>Transaction Class 1</td></tr>
<tr><td>0</td><td>Transaction Class 0</td></tr>
</table>
</td></tr>
<tr><td>0</td><td>Reserved. Transaction class 0 must always be mapped to VC0.</td></tr>
</table>

## 19.1.54 V0STS — Virtual Channel 0 Resource Status Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 11Ah–11Bh Attribute: RO
Default Value: 0000h Size: 16 bits

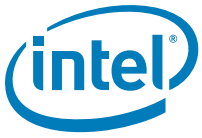
| Bit | Description |
|---|---|
| 15:2 | **Reserved**. |
| 1 | **VC Negotiation Pending (NP)** — RO.<br>0 = Negotiation is **not** pending.<br>1 = Virtual Channel is still being negotiated with ingress ports. |
| 0 | **Port Arbitration Tables Status (ATS)**. There is no port arbitration table for this VC; this bit is reserved as 0. |

## 19.1.55 UES — Uncorrectable Error Status Register (PCI Express—D28:F0/F1/F2/F3

Address Offset: 144h–147h Attribute: R/WC, RO
Default Value: 00000000000x0xxx0x0x0000000x0000bSize:32 bits

This register maintains its state through a platform reset. It loses its state upon suspend.

| Bit | Description |
|---|---|
| 31:21 | **Reserved** |
| 20 | **Unsupported Request Error Status (URE)** — R/WC.<br>0 = Unsupported request was **Not** received.<br>1 = Unsupported request was received. |
| 19 | **ECRC Error Status (EE)** — RO. ECRC is not supported. |



| Bit | Description |
|---|---|
| 18 | **Malformed TLP Status (MT)** — R/WC.<br>0 = Malformed TLP was **Not** received.<br>1 = Malformed TLP was received. |
| 17 | **Receiver Overflow Status (RO)** — R/WC.<br>0 = Receiver overflow did **Not** occur.<br>1 = Receiver overflow occurred. |
| 16 | **Unexpected Completion Status (UC)** — R/WC.<br>0 = Unexpected completion was **Not** received.<br>1 = Unexpected completion was received. |
| 15 | **Completion Abort Status (CA)** — R/WC.<br>0 = Completer abort was **Not** received.<br>1 = Completer abort was received. |
| 14 | **Completion Timeout Status (CT)** — R/WC.<br>0 = Completion did **Not** time out.<br>1 = Completion timed out. |
| 13 | Flow Control Protocol Error Status (FCPE) — RO. Flow Control Protocol Errors not supported. |
| 12 | **Poisoned TLP Status (PT)** — R/WC.<br>0 = Poisoned TLP was **Not** received.<br>1 = Poisoned TLP was received. |
| 11:5 | **Reserved** |
| 4 | **Data Link Protocol Error Status (DLPE)** — R/WC.<br>0 = Data link protocol error did **Not** occur.<br>1 = Data link protocol error occurred. |
| 3:1 | **Reserved** |
| 0 | **Training Error Status (TE)** — RO. Training Errors not supported. |

## 19.1.56 UEM — Uncorrectable Error Mask (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 148h–14Bh
Default Value: 00000000h
Attribute: R/WO, RO
Size: 32 bits

When set, the corresponding error in the UES register is masked, and the logged error will cause no action. When cleared, the corresponding error is enabled.

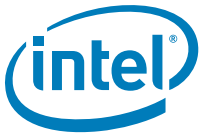
| Bit | Description |
|---|---|
| 31:21 | **Reserved** |
| 20 | **Unsupported Request Error Mask (URE)** — R/WO.<br>0 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is enabled.<br>1 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is masked. |
| 19 | ECRC Error Mask (EE) — RO. ECRC is not supported. |
| 18 | **Malformed TLP Mask (MT)** — R/WO.<br>0 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is enabled.<br>1 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is masked. |

| Bit | Description |
|---|---|
| 17 | **Receiver Overflow Mask (RO)** — R/WO.<br>0 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is enabled.<br>1 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is masked. |
| 16 | **Unexpected Completion Mask (UC)** — R/WO.<br>0 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is enabled.<br>1 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is masked. |
| 15 | Completion Abort Mask (CA) — R/WO.<br>0 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is enabled.<br>1 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is masked. |
| 14 | **Completion Timeout Mask (CT)** — R/WO.<br>0 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is enabled.<br>1 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is masked. |
| 13 | Flow Control Protocol Error Mask (FCPE) — RO. Flow Control Protocol Errors not supported. |
| 12 | **Poisoned TLP Mask (PT)** — R/WO.<br>0 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is enabled.<br>1 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is masked. |
| 11:5 | **Reserved** |
| 4 | **Data Link Protocol Error Mask (DLPE)** — R/WO.<br>0 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is enabled.<br>1 = The corresponding error in the UES register (D28:F0/F1/F2/F3:144) is masked. |
| 3:1 | **Reserved** |
| 0 | **Training Error Mask (TE)** — RO. Training Errors not supported |

## 19.1.57 UEV — Uncorrectable Error Severity (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 14Ch–14Fh
Default Value: 00060011h
Attribute: RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:21 | **Reserved** |
| 20 | **Unsupported Request Error Severity (URE)** — RO.<br>0 = Error considered non-fatal. (Default)<br>1 = Error is fatal. |
| 19 | ECRC Error Severity (EE) — RO. ECRC is not supported. |
| 18 | **Malformed TLP Severity (MT)** — RO.<br>0 = Error considered non-fatal.<br>1 = Error is fatal. (Default) |
| 17 | **Receiver Overflow Severity (RO)** — RO.<br>0 = Error considered non-fatal.<br>1 = Error is fatal. (Default) |
| 16 | **Unexpected Completion Severity (UC)** — RO.<br>0 = Error considered non-fatal. (Default)<br>1 = Error is fatal. |



| | |
|---|---|
| 15 | **Completion Abort Severity (CA)** — RO.<br>0 = Error considered non-fatal. (Default)<br>1 = Error is fatal. |
| 14 | **Completion Timeout Severity (CT)** — RO.<br>0 = Error considered non-fatal. (Default)<br>1 = Error is fatal. |
| 13 | Flow Control Protocol Error Severity (FCPE) — RO. Flow Control Protocol Errors not supported. |
| 12 | **Poisoned TLP Severity (PT)** — RO.<br>0 = Error considered non-fatal. (Default)<br>1 = Error is fatal. |
| 11:5 | **Reserved** |
| 4 | **Data Link Protocol Error Severity (DLPE)** — RO.<br>0 = Error considered non-fatal.<br>1 = Error is fatal. (Default) |
| 3:1 | **Reserved** |
| 0 | **Training Error Severity (TE)** — RO. TE is not supported. |

## 19.1.58 CES — Correctable Error Status Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 150h–153h Attribute: R/WC
Default Value: 00000000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:13 | **Reserved** |
| 12 | **Replay Timer Timeout Status (RTT)** — R/WC.<br>0 = Replay timer did **Not** time out.<br>1 = Replay timer timed out. |
| 11:9 | **Reserved** |
| 8 | **Replay Number Rollover Status (RNR)** — R/WC.<br>0 = Replay number did **Not** roll over.<br>1 = Replay number rolled over. |
| 7 | **Bad DLLP Status (BD)** — R/WC.<br>0 = Bad DLLP was **Not** received.<br>1 = Bad DLLP was received. |
| 6 | **Bad TLP Status (BT)** — R/WC.<br>0 = Bad TLP was **Not** received.<br>1 = Bad TLP was received. |
| 5:1 | **Reserved** |
| 0 | **Receiver Error Status (RE)** — R/WC.<br>0 = Receiver error did **Not** occur.<br>1 = Receiver error occurred. |

## 19.1.59 CEM — Correctable Error Mask Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 154h–157h
Default Value: 00000000h
Attribute: R/WO
Size: 32 bits

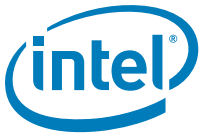
When set, the corresponding error in the CES register is masked, and the logged error will cause no action. When cleared, the corresponding error is enabled.

| Bit | Description |
|---|---|
| 31:13 | **Reserved** |
| 12 | **Replay Timer Timeout Mask (RTT)** — R/WO.<br>0 = **No** mask<br>1 = Mask for replay timer timeout. |
| 11:9 | **Reserved** |
| 8 | **Replay Number Rollover Mask (RNR)** — R/WO.<br>0 = **No** mask<br>1 = Mask for replay number rollover. |
| 7 | **Bad DLLP Mask (BD)** — R/WO.<br>0 = **No** mask<br>1 = Mask for bad DLLP reception. |
| 6 | **Bad TLP Mask (BT)** — R/WO.<br>0 = **No** mask<br>1 = Mask for bad TLP reception. |
| 5:1 | **Reserved** |
| 0 | **Receiver Error Mask (RE)** — R/WO.<br>0 = **No** mask<br>1 = Mask for receiver errors. |

## 19.1.60 AECC — Advanced Error Capabilities and Control Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 158h–15Bh
Default Value: 00000000h
Attribute: RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:9 | **Reserved** |
| 8 | **ECRC Check Enable (ECE)** — RO. ECRC is not supported. |
| 7 | **ECRC Check Capable (ECC)** — RO. ECRC is not supported. |
| 6 | **ECRC Generation Enable (EGE)** — RO. ECRC is not supported. |
| 5 | **ECRC Generation Capable (EGC)** — RO. ECRC is not supported. |
| 4:0 | **First Error Pointer (FEP)** — RO. |



## 19.1.61 RES — Root Error Status Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 170h–173h
Default Value: 00000000h
Attribute: R/WC, RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:27 | **Advanced Error Interrupt Message Number (AEMN)** — RO. There is only one error interrupt allocated. |
| 26:4 | **Reserved** |
| 3 | **Multiple ERR_FATAL/NONFATAL Received (MENR)** — RO. For Chipset, only one error will be captured. |
| 2 | **ERR_FATAL/NONFATAL Received (ENR)** — R/WC.<br>0 = **No** error message received.<br>1 = Either a fatal or a non-fatal error message is received. |
| 1 | **Multiple ERR_COR Received (MCR)** — RO. For Chipset, only one error will be captured. |
| 0 | **ERR_COR Received (CR)** — R/WC.<br>0 = **No** error message received.<br>1 = A correctable error message is received. |

## 19.1.62 RCTCL — Root Complex Topology Capability List Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 180–183h
Default Value: 00010005h
Attribute: RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:20 | **Next Capability (NEXT)** — RO. This field indicates the next item in the list, in this case, end of list. |
| 19:16 | **Capability Version (CV)** — RO. This field indicates the version of the capability structure. |
| 15:0 | **Capability ID (CID)** — RO. This field indicates this is a root complex topology capability. |

## 19.1.63 ESD — Element Self Description Register (PCI Express—D28:F0/F1/F2/F3)

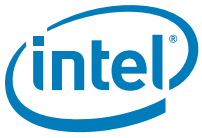
Address Offset: 184h–187h
Default Value: See Description
Attribute: RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:24 | **Port Number (PN)** — RO. This field indicates the ingress port number for the root port. There is a different value per port:<br>Port # / Value: 1 / 01h; 2 / 02h; 3 / 03h; 4 / 04h |
| 23:16 | **Component ID (CID)** — RO. This field returns the value of the ESD.CID field (Chipset Configuration Space: Offset 0104h:bits 23:16) of the chip configuration section, that is programmed by platform BIOS, since the root port is in the same component as the RCRB. |
| 15:8 | **Number of Link Entries (NLE)** — RO. (Default value is 01h). This field indicates one link entry (corresponding to the RCRB). |
| 7:4 | **Reserved**. |
| 3:0 | **Element Type (ET)** — RO. (Default value is 0h). This field indicates that the element type is a root port. |

## 19.1.64 ULD — Upstream Link Description Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 190h–193h
Default Value: 00000001h
Attribute: RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:24 | **Target Port Number (PN)** — RO. This field indicates the port number of the RCRB. |
| 23:16 | **Target Component ID (TCID)** — RO. This field returns the value of the ESD.CID field (Chipset Configuration Space: Offset 0104h:bits 23:16) of the chip configuration section, that is programmed by platform BIOS, since the root port is in the same component as the RCRB. |
| 15:2 | **Reserved**. |
| 1 | **Link Type (LT)** — RO. This bit indicates that the link points to the Chipset RCRB. |
| 0 | **Link Valid (LV)** — RO. This bit indicates that this link entry is valid. |



## 19.1.65 ULBA — Upstream Link Base Address Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 198h–19Fh Attribute: RO
Default Value: See Description Size: 64 bits

| Bit | Description |
|---|---|
| 63:32 | **Base Address Upper (BAU)** — RO. The RCRB of the Chipset is in 32-bit space. |
| 31:0 | **Base Address Lower (BAL)** — RO. This field matches the RCBA register (D31:F0:Offset F0h) value in the LPC bridge. |

## 19.1.66 PEETM — PCI Express Extended Test Mode Register (PCI Express—D28:F0/F1/F2/F3)

Address Offset: 318h Attribute: RO
Default Value: See Description Size: 8 bits

| Bit | Description |
|---|---|
| 7:3 | **Reserved** |
| 2 | **Scrambler Bypass Mode (BAU)** — R/W.<br>0 = Normal operation. Scrambler and descrambler are used.<br>1 = Bypasses the data scrambler in the transmit direction and the data de-scrambler in the receive direction.<br><br>**NOTE:** This functionality intended for debug/testing only.<br>**NOTE:** If bypassing scrambler with Chipset root port 1 in x4 configuration, each Chipset root port must have this bit set. |
| 1:0 | **Reserved** |

§

# 20 High Precision Event Timer Registers

The timer registers are memory-mapped in a non-indexed scheme. This allows the processor to directly access each register without having to use an index register. The timer register space is 1024 bytes. The registers are generally aligned on 64-bit boundaries to simplify implementation with IA64 processors. There are four possible memory address ranges beginning at 1) FED0_0000h, 2) FED0_1000h, 3) FED0_2000h., 4) FED0_4000h. The choice of address range will be selected by configuration bits in the High Precision Timer Configuration Register (Chipset Configuration Registers:Offset 3404h).

Behavioral Rules:

1. Software must not attempt to read or write across register boundaries. For example, a 32-bit access should be to offset x0h, x4h, x8h, or xCh. 32-bit accesses should not be to 01h, 02h, 03h, 05h, 06h, 07h, 09h, 0Ah, 0Bh, 0Dh, 0Eh, or 0Fh. Any accesses to these offsets will result in an unexpected behavior, and may result in a master abort. However, these accesses should not result in system hangs. 64-bit accesses can only be to x0h and must not cross 64-bit boundaries.
2. Software should not write to read-only registers.
3. Software should not expect any particular or consistent value when reading reserved registers or bits.

## 20.1 Memory Mapped Registers

**Table 20-149.Memory-Mapped Registers (Sheet 1 of 2)**

| Offset | Mnemonic | Register | Default | Type |
|---|---|---|---|---|
| 000–007h | GCAP_ID | General Capabilities and Identification | 0429B17F8 086A201h | RO |
| 008–00Fh | — | Reserved | — | — |
| 010–017h | GEN_CONF | General Configuration | 0000h | R/W |
| 018–01Fh | — | Reserved | — | — |
| 020–027h | GINTR_STA | General Interrupt Status | 00000000 00000000h | R/WC, R/W |
| 028–0EFh | — | **Reserved** | — | — |
| 0F0–0F7h | MAIN_CNT | Main Counter Value | N/A | R/W |
| 0F8–0FFh | — | **Reserved** | — | — |
| 100–107h | TIM0_CONF | Timer 0 Configuration and Capabilities | N/A | R/W, RO |
| 108–10Fh | TIM0_COMP | Timer 0 Comparator Value | N/A | R/W |

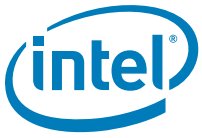


**Table 20-149.Memory-Mapped Registers (Sheet 2 of 2)**

| Offset | Mnemonic | Register | Default | Type |
|---|---|---|---|---|
| 110–11Fh | — | **Reserved** | — | — |
| 120–127h | TIM1_CONF | Timer 1 Configuration and Capabilities | N/A | R/W, RO |
| 128–12Fh | TIM1_COMP | Timer 1 Comparator Value | N/A | R/W |
| 130–13Fh | — | **Reserved** | — | — |
| 140–147h | TIM2_CONF | Timer 2 Configuration and Capabilities | N/A | R/W, RO |
| 148–14Fh | TIM2_COMP | Timer 2 Comparator Value | N/A | R/W |
| 150–15Fh | — | **Reserved** | — | — |
| 160–3FFh | — | **Reserved** | — | — |

**NOTES:**
1. Reads to reserved registers or bits will return a value of 0.
2. Software must not attempt locks to the memory-mapped I/O ranges for High Precision Event Timers. If attempted, the lock is not honored, which means potential deadlock conditions may occur.

## 20.1.1 GCAP_ID—General Capabilities and Identification Register

Address Offset: 00h Attribute: RO
Default Value: 0429B17F8086A201h Size: 64 bits

| Bit | Description |
|---|---|
| 63:32 | **Main Counter Tick Period (COUNTER_CLK_PER_CAP)** — RO. This field indicates the period at which the counter increments in femptoseconds (10^-15 seconds). This will return 0429B17Fh when read. This indicates a period of 69841279h fs (69.841279 ns). |
| 31:16 | **Vendor ID Capability (VENDOR_ID_CAP)** — RO. This is a 16-bit value assigned to Intel. |
| 15 | **Legacy Replacement Rout Capable (LEG_RT_CAP)** — RO. Hardwired to 1. Legacy Replacement Interrupt Rout option is supported. |
| 14 | **Reserved**. This bit returns 0 when read. |
| 13 | **Counter Size Capability (COUNT_SIZE_CAP)** — RO. Hardwired to 1. Counter is 64-bit wide. |
| 12:8 | **Number of Timer Capability (NUM_TIM_CAP)** — RO. This field indicates the number of timers in this block.<br>02h = Three timers. |
| 7:0 | **Revision Identification (REV_ID)** — RO. This indicates which revision of the function is implemented. Default value will be 01h. |

### 20.1.2 GEN_CONF—General Configuration Register

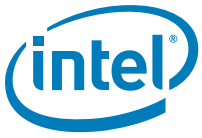
Address Offset: 010h Attribute: R/W
Default Value: 00000000 00000000h Size: 64 bits

| Bit | Description |
|---|---|
| 63:2 | **Reserved**. These bits return 0 when read. |
| 1 | **Legacy Replacement Rout (LEG_RT_CNF)** — R/W. If the ENABLE_CNF bit and the LEG_RT_CNF bit are both set, then the interrupts will be routed as follows:<br>• Timer 0 is routed to IRQ0 in 8259 or IRQ2 in the I/O APIC<br>• Timer 1 is routed to IRQ8 in 8259 or IRQ8 in the I/O APIC<br>• Timer 2-n is routed as per the routing in the timer n config registers.<br>• If the Legacy Replacement Rout bit is set, the individual routing bits for Timers 0 and 1 (APIC) will have no impact.<br>• If the Legacy Replacement Rout bit is not set, the individual routing bits for each of the timers are used.<br>• This bit will default to 0. BIOS can set it to 1 to enable the legacy replacement routing, or 0 to disable the legacy replacement routing. |
| 0 | **Overall Enable (ENABLE_CNF)** — R/W. This bit must be set to enable any of the timers to generate interrupts.<br>0 = Disable. The main counter will halt (will not increment) and no interrupts will be caused by any of these timers. For level-triggered interrupts, if an interrupt is pending when the ENABLE_CNF bit is changed from 1-to-0, the interrupt status indications (in the various Txx_INT_STS bits) will not be cleared. Software must write to the Txx_INT_STS bits to clear the interrupts.<br>1 = Enable.<br><br>**NOTE:** This bit will default to 0. BIOS can set it to 1 or 0. |

### 20.1.3 GINTR_STA—General Interrupt Status Register

Address Offset: 020h Attribute: R/W, R/WC
Default Value: 00000000 00000000h Size: 64 bits

| Bit | Description |
|---|---|
| 63:3 | **Reserved**. These bits will return 0 when read. |
| 2 | **Timer 2 Interrupt Active (T02_INT_STS)** — R/W. Same functionality as Timer 0. |
| 1 | **Timer 1 Interrupt Active (T01_INT_STS)** — R/W. Same functionality as Timer 0. |
| 0 | **Timer 0 Interrupt Active (T00_INT_STS)** — R/WC. The functionality of this bit depends on whether the edge or level-triggered mode is used for this timer. (default = 0)<br>If set to level-triggered mode:<br>This bit will be set by hardware if the corresponding timer interrupt is active. Once the bit is set, it can be cleared by software writing a 1 to the same bit position. Writes of 0 to this bit will have no effect.<br>If set to edge-triggered mode:<br>This bit should be ignored by software. Software should always write 0 to this bit.<br><br>**NOTE:** Defaults to 0. In edge triggered mode, this bit will always read as 0 and writes will have no effect. |



### 20.1.4 MAIN_CNT—Main Counter Value Register

Address Offset: 0F0h
Default Value: N/A
Attribute: R/W
Size: 64 bits

| Bit | Description |
|---|---|
| 63:0 | **Counter Value (COUNTER_VAL[63:0])** — R/W. Reads return the current value of the counter. Writes load the new value to the counter.<br><br>**NOTES:**<br>1. Writes to this register should only be done while the counter is halted.<br>2. Reads to this register return the current value of the main counter.<br>3. 32-bit counters will always return 0 for the upper 32-bits of this register.<br>4. If 32-bit software attempts to read a 64-bit counter, it should first halt the counter. Since this delays the interrupts for all of the timers, this should be done only if the consequences are understood. It is strongly recommended that 32-bit software only operate the timer in 32-bit mode.<br>5. Reads to this register are monotonic. No two consecutive reads return the same value. The second of two reads always returns a larger value (unless the timer has rolled over to 0). |

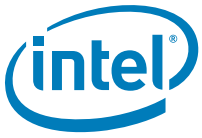
### 20.1.5 TIMn_CONF—Timer n Configuration and Capabilities Register

Address Offset: Timer 0: 100–107h, Timer 1: 120–127h, Timer 2: 140–147h
Default Value: N/A
Attribute: RO, R/W
Size: 64 bits

***Note:*** The letter n can be 0, 1, or 2, referring to Timer 0, 1 or 2.

| Bit | Description |
|---|---|
| 63:56 | **Reserved**. These bits will return 0 when read. |
| 55:52, 43 | **Timer Interrupt Rout Capability (TIMERn_INT_ROUT_CAP)** — RO.<br>Timer 0, 1:Bits 52, 53, 54, and 55 in this field (corresponding to IRQ 20, 21, 22, and 23) have a value of 1. Writes will have no effect.<br>Timer 2:Bits 43, 52, 53, 54, and 55 in this field (corresponding to IRQ 11, 20, 21, 22, and 23) have a value of 1. Writes will have no effect.<br><br>**NOTE:** If IRQ 11 is used for HPET #2, software should ensure IRQ 11 is not shared with any other devices to ensure the proper operation of HPET #2. |
| 51:44, 42:14 | **Reserved.** These bits return 0 when read. |

| Bit | Description |
|---|---|
| 13:9 | **Interrupt Rout (TIMERn_INT_ROUT_CNF)** — R/W. This 5-bit field indicates the routing for the interrupt to the I/O (x) APIC. Software writes to this field to select which interrupt in the I/O (x) will be used for this timer's interrupt. If the value is not supported by this particular timer, then the value read back will not match what is written. The software must only write valid values.<br><br>**NOTES:**<br>1. If the Legacy Replacement Rout bit is set, then Timers 0 and 1 will have a different routing, and this bit field has no effect for those two timers.<br>2. Timer 0,1: Software is responsible to make sure it programs a valid value (20, 21, 22, or 23) for this field. The Chipset logic does not check the validity of the value written.<br>3. Timer 2: Software is responsible to make sure it programs a valid value (11, 20, 21, 22, or 23) for this field. The Chipset logic does not check the validity of the value written. |
| 8 | **Timer n 32-bit Mode (TIMERn_32MODE_CNF)** — R/W or RO. Software can set this bit to force a 64-bit timer to behave as a 32-bit timer. This is typically needed if software is not willing to halt the main counter to read or write a particular timer, and the software is not capable of atomic 64-bit operations to the timer. This bit is only relevant if the timer is operating in 64-bit mode in which case that timer can be forced to 32-bit mode by setting this bit. When Timer 0 is switched to 32-bit mode, the upper 32-bits are loaded with 0's which will remain when the timer is switched back to 64-bit mode. If the timer is not in 64-bit mode, then this bit will always be read as 0 and writes will have no effect.<br>Timer 0:Bit is read/write (default to 0). 0 = 64 bit; 1= 32 bit<br>Timers 1, 2:Hardwired to 0. Writes have no effect since these timers are 32-bit only. |
| 7 | **Reserved**. This bit returns 0 when read. |
| 6 | **Timer n Value Set (TIMERn_VAL_SET_CNF)** — R/W. Software uses this bit only for Timer 0 if it has been set to periodic mode. By writing this bit to a 1, the software is then allowed to directly set the timer's accumulator. Software does **not** have to write this bit back to 1 (it automatically clears).<br>Software should not write a 1 to this bit position if the timer is set to non-periodic mode.<br><br>**NOTE:** This bit will return 0 when read. Writes will only have an effect for Timer 0 if it is set to periodic mode. Writes will have no effect for Timers 1 and 2. |
| 5 | **Timer n Size (TIMERn_SIZE_CAP)** — RO. This read only field indicates the size of the timer.<br>Timer 0:Value is 1 (64-bits).<br>Timers 1, 2:Value is 0 (32-bits). |
| 4 | **Periodic Interrupt Capable (TIMERn_PER_INT_CAP)** — RO. If this bit is 1, the hardware supports a periodic mode for this timer's interrupt.<br>Timer 0: Hardwired to 1 (supports the periodic interrupt).<br>Timers 1, 2: Hardwired to 0 (does not support periodic interrupt). |
| 3 | **Timer n Type (TIMERn_TYPE_CNF)** — R/W or RO.<br>Timer 0:Bit is read/write. 0 = Disable timer to generate periodic interrupt; 1 = Enable timer to generate a periodic interrupt.<br>Timers 1, 2: Hardwired to 0. Writes have no effect. |
| 2 | **Timer n Interrupt Enable (TIMERn_INT_ENB_CNF)** — R/W. This bit must be set to enable timer n to cause an interrupt when it times out.<br>0 = Enable.<br>1 = Disable (Default). The timer can still count and generate appropriate status bits, but will not cause an interrupt. |



| Bit | Description |
|---|---|
| 1 | **Timer Interrupt Type (TIMERn_INT_TYPE_CNF)** — R/W.<br>0 = The timer interrupt is edge triggered. This means that an edge-type interrupt is generated. If another interrupt occurs, another edge will be generated.<br>1 = The timer interrupt is level triggered. This means that a level-triggered interrupt is generated. The interrupt will be held active until it is cleared by writing to the bit in the General Interrupt Status Register. If another interrupt occurs before the interrupt is cleared, the interrupt will remain active. |
| 0 | **Reserved.** These bits will return 0 when read. |

**NOTE:** Reads or writes to unimplemented timers should not be attempted. Read from any unimplemented registers will return an undetermined value.

## 20.1.6 TIMn_COMP—Timer n Comparator Value Register

Address Offset: Timer 0: 108h–10Fh,
Timer 1: 128h–12Fh,
Timer 2: 148h–14Fh
Attribute: R/W
Default Value: N/A Size: 64 bit

| Bit | Description |
|---|---|
| 63:0 | **Timer Compare Value** — R/W. Reads to this register return the current value of the comparator<br>Timers 0, 1, or 2 are configured to non-periodic mode:<br>Writes to this register load the value against which the main counter should be compared for this timer.<br>• When the main counter equals the value last written to this register, the corresponding interrupt can be generated (if so enabled).<br>• The value in this register does not change based on the interrupt being generated.<br>Timer 0 is configured to periodic mode:<br>• When the main counter equals the value last written to this register, the corresponding interrupt can be generated (if so enabled).<br>• After the main counter equals the value in this register, the value in this register is increased by the value last written to the register.<br><br>For example, if the value written to the register is 00000123h, then<br><br>1. An interrupt will be generated when the main counter reaches 00000123h.<br>2. The value in this register will then be adjusted by the hardware to 00000246h.<br>3. Another interrupt will be generated when the main counter reaches 00000246h<br>4. The value in this register will then be adjusted by the hardware to 00000369h<br><br>• As each periodic interrupt occurs, the value in this register will increment. When the incremented value is greater than the maximum value possible for this register (FFFFFFFFh for a 32-bit timer or FFFFFFFFFFFFFFFFh for a 64-bit timer), the value will wrap around through 0. For example, if the current value in a 32-bit timer is FFFF0000h and the last value written to this register is 20000, then after the next interrupt the value will change to 00010000h<br>Default value for each timer is all 1s for the bits that are implemented. For example, a 32-bit timer has a default value of 00000000FFFFFFFFh. A 64-bit timer has a default value of FFFFFFFFFFFFFFFFh. |

§

# 21 Serial Peripheral Interface (SPI)

The Serial Peripheral Interface resides in memory mapped space. This function contains registers that allow for the setup and programming of devices that reside on the SPI interface.

***Note:*** All registers in this function (including memory-mapped registers) must be addressable in Byte, Word, and DWord quantities. The software must always make register accesses on natural boundaries (i.e., DWord accesses must be on DWord boundaries; word accesses on word boundaries, etc.) In addition, the memory-mapped register space must not be accessed with the LOCK semantic exclusive-access mechanism. If software attempts exclusive-access mechanisms to the SPI memory-mapped space, the results are undefined.

## 21.1 Serial Peripheral Interface Memory Mapped Configuration Registers

The SPI Host Interface registers are memory-mapped in the RCRB Chipset Register Space with a base address (SPIBAR) of 3020h and are located within the range of 3020h to 308Fh. The individual registers are then accessible at SPIBAR + Offset as indicated in the following table.

These memory mapped registers must be accessed in byte, word, or DWord quantities.

**Table 21-150.Serial Peripheral Interface (SPI) Register Address Map (SPI Memory Mapped Configuration Registers) (Sheet 1 of 2)**

| SPIBAR + Offset | Mnemonic | Register Name | Default | Access |
|---|---|---|---|---|
| 00h–01h | SPIS | SPI Status | See Register Description | RO, R/WC, R/WLO |
| 02h–03h | SPIC | SPI Control | 0001h | R/W |
| 04h–07h | SPIA | SPI Address | 00000000h | R/W |
| 08h–0Fh | SPID0 | SPI Data 0 | See Register Description | R/W |
| 10h–17h | SPID1 | SPI Data 1 | 00000000 00000000h | R/W |
| 18h–1Fh | SPID2 | SPI Data 2 | 00000000 00000000h | R/W |
| 20h–27h | SPID3 | SPI Data 3 | 00000000 00000000h | R/W |
| 28h–2Fh | SPID4 | SPI Data 4 | 00000000 00000000h | R/W |
| 30h–37h | SPID5 | SPI Data 5 | 00000000 00000000h | R/W |

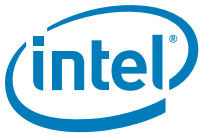


**Table 21-150.Serial Peripheral Interface (SPI) Register Address Map (SPI Memory Mapped Configuration Registers) (Sheet 2 of 2)**

| SPIBAR + Offset | Mnemonic | Register Name | Default | Access |
|---|---|---|---|---|
| 38h–3Fh | SPID6 | SPI Data 6 | 00000000 00000000h | R/W |
| 40h–47h | SPID7 | SPI Data 7 | 00000000 00000000h | R/W |
| 50h–53h | BBAR | BIOS Base Address Configuration | 00000000h | R/W |
| 54h–55h | PREOP | Prefix Opcode Configuration | 0004h | R/W |
| 56h–57h | OPTYPE | Opcode Type Configuration | 0000h | R/W |
| 58h–5Fh | OPMENU | Opcode Menu Configuration | 00000000 00000005h | R/W |
| 60h–63h | PBR0 | Protected BIOS Range 0 | 00000000h | R/W |
| 64h–67h | PBR1 | Protected BIOS Range 1 | 00000000h | R/W |
| 68h–6Bh | PBR2 | Protected BIOS Range 2 | 00000000h | R/W |
| 6Ch–6Fh | — | **Reserved** | — | — |

## 21.1.1 SPIS—SPI Status Register (SPI Memory Mapped Configuration Registers)

Memory Address:SPIBAR + 00h Attribute: RO, R/WC, R/WLO

Default Value: See bit descriptionSize:16 bits

| Bit | Description |
|---|---|
| 15 | **SPI Configuration Lock-Down** — R/WLO.<br>0 = **No** Lock-Down (Default)<br>1 = SPI Static Configuration information in offsets 50h through 6Fh can not be overwritten. Once set to 1, this bit can only be cleared by a hardware reset. |
| 14:4 | **Reserved** |
| 3 | **Blocked Access Status** — R/WC.<br>0 = **Not** blocked (Default)<br>1 = Hardware sets this bit to 1 when an access is blocked from running on the SPI interface due to one of the protection policies or when any of the programmed cycle registers is written while a programmed access is already in progress. This bit is set for both programmed accesses and direct memory reads that get blocked.<br><br>**NOTE:** This bit remains asserted until cleared by software writing a 1 or hardware reset. |
| 2 | **Cycle Done Status**— R/WC.<br>0 = **Not** done (Default)<br>1 = The Chipset sets this bit to 1 when the SPI Cycle completes (i.e., SCIP bit is 0) after software sets the SCGO bit.<br><br>**NOTE:** This bit remains asserted until cleared by software writing a 1 or hardware reset.<br>**NOTE:** Software must make sure this bit is cleared prior to enabling the SPI SMI# assertion for a new programmed access.<br>**NOTE:** This bit gets set after the Status Register Polling sequence completes after reset deasserts. It is cleared before and during that sequence. |

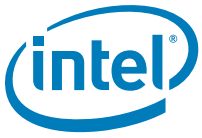
| Bit | Description |
|---|---|
| 1 | **SPI Access Grant** — RO. This bit is used by the software to know when the other SPI master will not be initiating any long transactions on the SPI bus.<br>0 = Default<br>1 = It is set by hardware in response to software setting the SPI Access Request bit and completing the Future Pending handshake with the LAN component.<br><br>**NOTE:** This bit is cleared in response to software clearing the SPI Access Request bit. |
| 0 | **SPI Cycle In Progress (SCIP)** — RO.<br>0 = Cycle **Not** in Progress (Default)<br>1 = Hardware sets this bit when software sets the SPI Cycle Go bit in the Command register. This bit remains set until the cycle completes on the SPI interface. Hardware automatically sets and clears this bit so that software can determine when read data is valid and/or when it is safe to begin programming the next command.<br>This bit reports 1b during the Status Register Polling sequence after reset deasserts; it is cleared when that sequence completes.<br><br>**NOTE:** Software must only program the next command when this bit is 0. |

### 21.1.2 SPIC—SPI Control Register (SPI Memory Mapped Configuration Registers)

Memory Address:SPIBAR + 02h Attribute: R/W

Default Value: 4005hSize:16 bits

| Bit | Description |
|---|---|
| 15 | **SPI SMI# Enable** — R/W.<br>0 = Disable.<br>1 = Enable. The SPI asserts an SMI# request when the Cycle Done Status bit is 1. |
| 14 | **DATA Cycle**— R/W.<br>0 = No data is delivered for this cycle, and the DBC and data fields themselves are don't cares.<br>1 = There is data that corresponds to this transaction. |
| 13:8 | **Data Byte Count (DBC) —** R/W. This field specifies the number of bytes to shift in or out during the data portion of the SPI cycle. The valid settings (in decimal) are any value from 0 to 63. The number of bytes transferred is the value of this field plus 1.<br>For example, when this field is 000000b, then there is 1 byte to transfer and that 111111b means there are 64 bytes to transfer. |
| 7 | **Reserved** |
| 6:4 | **Cycle Opcode Pointer** — R/W. This field selects one of the programmed opcodes in the Opcode Menu to be used as the SPI Command/Opcode. In the case of an Atomic Cycle Sequence, this determines the second command. |
| 3 | **Sequence Prefix Opcode Pointer** — R/W. This field selects one of the two programmed prefix opcodes for use when performing an Atomic Cycle Sequence. By making this programmable, the Chipset supports flash devices that have different opcodes for enabling writes to the data space vs. status register<br>0 = A value of 0 points to the opcode in the least significant byte of the Prefix Opcodes register. |



| | |
|---|---|
| 2 | **Atomic Cycle Sequence (ACS)** — R/W.<br>0 = **No** atomic cycle sequence.<br>1 = When set to 1 along with the SCGO assertion, the Chipset will execute a sequence of commands on the SPI interface without allowing the LAN component to arbitrate and interleave cycles. |
| 1 | **SPI Cycle Go (SCGO) —** R/W. This bit always returns 0 on reads.<br>0 = SPI cycle **Not** started.<br>1 = A write to this register with a 1 in this bit starts the SPI cycle defined by the other bits of this register. The “SPI Cycle in Progress” (SCIP) bit gets set by this action.<br><br>**NOTE:** Writes to this bit while the Cycle In Progress bit is set are ignored.<br>**NOTE:** Other bits in this register can be programmed for the same transaction when writing this bit to 1. |
| 0 | **SPI Access Request** — R/W. This bit is used by software to request that the other SPI master stop initiating long transactions on the SPI bus.<br>0 = **No** request.<br>1 = Request that the other SPI master stop initiating long transactions on the SPI bus.<br><br>**NOTE:** This bit defaults to a 1 and must be cleared by BIOS after completing the accesses for the boot process. |

### 21.1.3 SPIA—SPI Address Register (SPI Memory Mapped Configuration Registers)

Memory Address:SPIBAR + 04h Attribute: R/W

Default Value: 00000000hSize:32 bits

| Bit | Description |
|---|---|
| 31:24 | **Reserved** |
| 23:0 | **SPI Cycle Address (SCA)** — R/W. This field is shifted out as the SPI Address (MSb first). Bits 23:0 correspond to Address bits 23:0. |

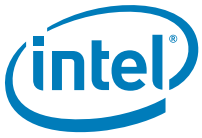
### 21.1.4 SPID[N] —SPI Data N Register (SPI Memory Mapped Configuration Registers)

Memory Address: SPI Data [0]: SPIBAR + 08h Attribute: R/W
SPI Data [1]: SPIBAR + 10h
SPI Data [2]: SPIBAR + 18h
SPI Data [3]: SPIBAR + 20h
SPI Data [4]: SPIBAR + 28h
SPI Data [5]: SPIBAR + 30h
SPI Data [6]: SPIBAR + 38h
SPI Data [7]: SPIBAR + 40h

Default Value: See Notes 1 and 2 below Size: 64 bits

| Bit | Description |
|---|---|
| 63:0 | **SPI Cycle Data [N] (SCD[N])** — R/W. This field is shifted out as the SPI Data on the Master-Out Slave-In Data pin during the data portion of the SPI cycle. The SCD[N] register does not begin shifting until SPID[N-1] has completely shifted in/out.<br>This register also shifts in the data from the Master-In Slave-Out pin into this register during the data portion of the SPI cycle.<br><br>**NOTE:** The data is always shifted starting with the least significant byte, msb to lsb, followed by the next least significant byte, msb to lsb, etc. Specifically, the shift order on SPI in terms of bits within this register is: 7-6-5-4-3-2-1-0-15-14-13-…8-23-22-…16-31…24-39..32…etc. Bit 56 is the last bit shifted out/in. There are no alignment assumptions; byte 0 always represents the value specified by the cycle address.<br>**NOTE:** The data in this register may be modified by the hardware during any programmed SPI transaction. Direct Memory Reads do not modify the contents of this register. |

**NOTES:**
1. For SPI Data [7:1] Registers Only: Default value is 0000000000000000h.
2. For SPI Data 0 Register default value only: This register is initialized to 0 by the reset assertion. However, the least significant byte of this register is loaded with the first Status Register read of the Atomic Cycle Sequence that the hardware automatically runs out of reset. Therefore, bit 0 of this register can be read later to determine if the platform encountered the boundary case in which the SPI flash was busy with an internal instruction when the platform reset deasserted.



### 21.1.5 BBAR—BIOS Base Address Register (SPI Memory Mapped Configuration Registers)

Memory Address:SPIBAR + 50h Attribute: R/W

Default Value: 00000000hSize:32 bits

| Bit | Description |
|---|---|
| 31:24 | **Reserved**. |
| 23:8 | **Bottom of System Flash** — R/W. This field determines the bottom of the System BIOS. The Chipset will not run programmed commands nor memory reads whose address field is less than this value. This field corresponds to bits 23:8 of the 3-byte address; bits 7:0 are assumed to be 00h for this vector when comparing to a potential SPI address.<br><br>**NOTE:** Software must always program 1s into the upper, Don't Care, bits of this field based on the flash size. Hardware does not know the size of the flash array and relies upon the correct programming by software. The default value of 0000h results in all cycles allowed.<br>**NOTE:** In the event that this value is programmed below some of the BIOS Memory segments, described above, this protection policy takes precedence. |
| 7:0 | **Reserved** |

**NOTE:** This register is not writable when the SPI Configuration Lock-Down bit (SPIBAR + 00h:15) is set.

### 21.1.6 PREOP—Prefix Opcode Configuration Register (SPI Memory Mapped Configuration Registers)

Memory Address:SPIBAR + 54h Attribute: R/W

Default Value: 0004hSize:16 bits

| Bit | Description |
|---|---|
| 15:8 | **Prefix Opcode 1**— R/W. Software programs an SPI opcode into this field that is permitted to run as the first command in an atomic cycle sequence. |
| 7:0 | **Prefix Opcode 0** — R/W. Software programs an SPI opcode into this field that is permitted to run as the first command in an atomic cycle sequence. |

**NOTE:** This register is not writable when the SPI Configuration Lock-Down bit (SPIBAR + 00h:15) is set.

### 21.1.7 OPTYPE—Opcode Type Configuration Register (SPI Memory Mapped Configuration Registers)

Memory Address:SPIBAR + 56h Attribute: R/W

Default Value: 0000hSize:16 bits

Entries in this register correspond to the entries in the Opcode Menu Configuration register.

***Note:*** The definition below only provides write protection for opcodes that have addresses associated with them. Therefore, any erase or write opcodes that do not use an address should be avoided (for example, “Chip Erase” and “Auto-Address Increment Byte Program”)

| Bit | Description |
|---|---|
| 15:14 | **Opcode Type 7** — R/W. See the description for bits 1:0 |
| 13:12 | **Opcode Type 6** — R/W. See the description for bits 1:0 |
| 11:10 | **Opcode Type 5** — R/W. See the description for bits 1:0 |
| 9:8 | **Opcode Type 4** — R/W. See the description for bits 1:0 |
| 7:6 | **Opcode Type 3** — R/W. See the description for bits 1:0 |
| 5:4 | **Opcode Type 2** — R/W. See the description for bits 1:0 |
| 3:2 | **Opcode Type 1** — R/W. See the description for bits 1:0 |
| 1:0 | **Opcode Type 0** — R/W. This field specifies information about the corresponding Opcode 0. This information allows the hardware to, 1) know whether to use the address field and, 2) provide BIOS and Shared Flash protection capabilities. The encoding of the two bits is:<br>00 = No address associated with this Opcode; Read cycle type<br>01 = No address associated with this Opcode; Write cycle type<br>10 = Address required; Read cycle type<br>11 = Address required; Write cycle type |

**NOTE:** This register is not writable when the SPI Configuration Lock-Down bit (SPIBAR + 00h:15) is set.

### 21.1.8 OPMENU—Opcode Menu Configuration Register (SPI Memory Mapped Configuration Registers)

Memory Address:SPIBAR + 58h Attribute: R/W

Default Value: 0000000000000005hSize:64 bits

Eight entries are available in this register to give BIOS a sufficient set of commands for communicating with the flash device, while also restricting what malicious software can do. This keeps the hardware flexible enough to operate with a wide variety of SPI devices.

***Note:*** It is recommended that BIOS avoid programming Write Enable opcodes in this menu. Malicious software could then perform writes and erases to the SPI flash without using the atomic cycle mechanism. This could cause functional failures in a shared flash

environment. Write Enable opcodes should only be programmed in the Prefix Opcodes.

| Bit | Description |
|---|---|
| 63:56 | **Allowable Opcode 7** — R/W. See the description for bits 7:0 |
| 55:48 | **Allowable Opcode 6** — R/W. See the description for bits 7:0 |
| 47:40 | **Allowable Opcode 5** — R/W. See the description for bits 7:0 |
| 39:32 | **Allowable Opcode 4** — R/W. See the description for bits 7:0 |
| 31:24 | **Allowable Opcode 3** — R/W. See the description for bits 7:0 |
| 23:16 | **Allowable Opcode 2** — R/W. See the description for bits 7:0 |
| 15:8 | **Allowable Opcode 1** — R/W. See the description for bits 7:0 |
| 7:0 | **Allowable Opcode 0** — R/W. Software programs an SPI opcode into this field for use when initiating SPI commands through the Control Register. |

**NOTE:** This register is not writable when the SPI Configuration Lock-Down bit (SPIBAR + 00h:15) is set.

### 21.1.9 PBR[N]—Protected BIOS Range [N] (SPI Memory Mapped Configuration Registers)

Memory Address: PBR[0]: SPIBAR + 60h  Attribute: R/W
PBR[1]: SPIBAR + 64h
PBR[2]: SPIBAR + 68h

Default Value: 00000000h  Size: 32 bits

| Bit | Description |
|---|---|
| 31 | **Write Protection Enable** — R/W.<br>0 = Disable. The base and limit fields are ignored when this bit is cleared.<br>1 = Enable. The Base and Limit fields in this register are valid. |
| 30:24 | **Reserved** |
| 23:12 | **Protected Range Limit** — R/W. This field corresponds to SPI address bits 23:12 and specifies the upper limit of the protected range.<br><br>**NOTE:** Any address greater than the value programmed in this field is unaffected by this protected range. |
| 11:0 | **Protected Range Base** — R/W. This field corresponds to SPI address bits 23:12 and specifies the lower base of the protected range.<br><br>**NOTE:** Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range. |

**NOTE:** This register is not writable when the SPI Configuration Lock-Down bit (SPIBAR + 00h:15) is set.

§